September 1982 Volume 1 $2.00

softalk

for the IBM Personal Computer

Kenneth Bowles, Standard Bearer

Exec SofTech

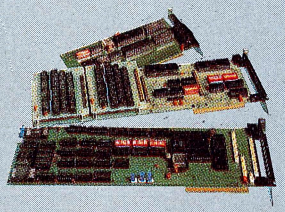
Pascal vs. p-System [II]

First, the IBM Personal Computer,
The Next Step . . .

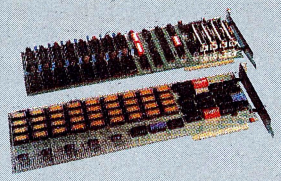
Tecmar

The TECMAR Expansion series is the first and only, complete line of expansion options available for the IBM Personal Computer.

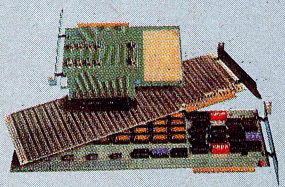
Now totaling over twenty-five separate options, the TECMAR series gives you the broadest range of expansion available for your IBM Personal Computer.



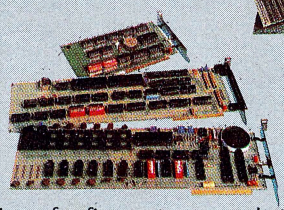
SYSTEM EXPANSION with a complete Expansion Chassis providing six additional system slots, a separate power supply and styling that complements the IBM system.



MEMORY EXPANSION in 64K 128K, 192K and 256K Byte increments of Dynamic RAM with parity. 32K Bytes of Static RAM, 32K Bytes of CMOS RAM with battery backup, or up to 128K Bytes of Read Only Memory.



PRACTICAL EXPANSION with two Serial ports and one Parallel port on a single board, or a Time of Day calendar with battery backup, a Voice Synthesizer with vocabulary in ROM and phoneme speech generation, even a BSR X10™ device controller for lights and appliances.



DISK EXPANSION through the addition of a five or ten megabyte Winchester disk. The disk options come enclosed in the TECMAR Expansion Chassis, providing additional expansion slots as well as Winchester disk storage. This approach assures you of unmatched system expandability for nearly any application.

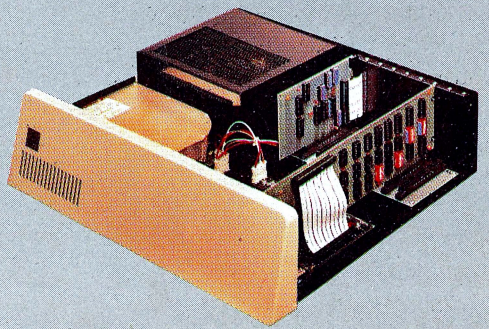
FUNCTIONAL EXPANSION is also available with TECMAR Speed Disk™ and print Spooling Software that give new functionality to memory options.

UNMATCHED EXPANSION for the serious IBM Personal Computer user through these and the many other TECMAR Expansion products available through participating COMPUTERLAND stores, and other fine computer retailers nationwide.

LABORATORY/INDUSTRIAL EXPANSION through an IEEE 488 interface, the Lab Tender with an 8 bit A/D and D/A, or the Lab Master for 12 bit A/D and D/A, a two axis Stepper Motor Controller, or the Parallel Digital-In/Digital-Out Base Board™, Video Digitization with Video VanGogh™.

DEVELOPMENT EXPANSION using an E+EEPROM programmer, Protozoa prototyping boards or a TECMAR Extender card.

NEW PRODUCTS are currently under development with many soon to be announced. At present shipping 26 unique IBM add-on products, we are still looking for needs to meet. If you have an need for a new product for the IBM Personal Computer, and would like to ask us about it, give a call on our Product Input Hotline at (216) 464-8317.



For IBM Personal Computer Expansion, TAKE THE NEXT STEP . . .

Tecmar Inc.

PERSONAL COMPUTER PRODUCTS DIVISION
23600 Mercantile Road, Cleveland, Ohio 44122
Telephone: (216) 464-7410 Telex: 241735

softalk

for the IBM Personal Computer

Features

Beginners' Corner

How to manipulate files, using the commands *load*, *run*, *list*, and others.

Craig Stinson 4

Profit Plot

Using spreadsheets to estimate your retirement income, allowing for inflation.

Shirley Fisher 11

Exec SofTech

The marriage of Ken Bowles's p-System and Al Irvine's business expertise changed the face of the software industry.

Al Tommervik 18

Boards and Buses

How to choose a printer: parallel versus serial, and dot-matrix versus typewriter quality.

Richard Kaapke 24

Kenneth Bowles: The Prophet of Portability

When Kenneth Bowles lost his job as director of the UCSD computer center, it was not the end, but the beginning.

Al Tommervik 26

IBM Pascal versus the p-System (Part 2)

The conclusion of a two-part series comparing the versions of Pascal.

Bruce F. Webster and Deirdre A. Wendt 34

Patching Pascal

IBM Pascal not performing as it should? Here's a patch from the manufacturer 38

System Notebook

How DOS perceives time; transferring DOS; using DOS to compare disks.

Alan Boyd 43

Departments

Crosstalk 2
Tradetalk 17
Marketalk Reviews 30
Newspeak 40
Marketalk News 47
Bestsellers 51

In Next Month's Softalk

Tying Personal Computers together to create an office network . . . a critical comparison of PC-DOS and CP/M-86 . . . Exec Infocom: What's new from the creators of *Zork* . . . the first article in a tutorial series on Pascal . . . and more.

Index of Advertisers

Alpha Software 6, 42
Amdek 3
Beaman Porter 15
Compu-haus 38
Continental Software ..... 5, 51, Cover 4
Corona Data Systems ..... Cover 3
DataMost 16
Davidson & Associates 45
Emerging Technology Consultants 21
Ensign Software 33
Howard Software Services 10
Information Unlimited Software 50
Insoft 9
Interface Inc. 36
International Computers 12
Laboratory Microsystems 32
Max Software Publishing 13
NF Systems 8
Peter Norton 41
Omric 46
Pascal Systems 37
PC Connection 52
Performance Engineered Programming 20
PrintaColor 29
Quadram Corp. 22-23
Quentin Research 14
Softalk for the IBM Personal Computer 17
Software Options 44
SSR Corporation 49
Synergistic Software 39
Tecmar Inc. ..... Cover 2
Yucaipa Software 11

Softalk Publishing Inc.

Chairman	John Haller
Publisher	Al Tommervik
Editor-in-Chief	Margot Comstock Tommervik
Associate Publisher	Mary Sue Rennells
Accounting	Evelyn Burke
Editorial Manager	Patricia Ryall

Softalk for the IBM Personal Computer

Editor	Craig Stinson
Art Director	Kurt A. Wahlner
Managing Editor	Forrest Johnson
Newspeak Editor	David Hunter
Guest Reviewers	Bill Pittore Bob Trammel
Proofreader	Harry McNeil
Contributing Editors	
Financial Modeling	Shirley Fisher
Hardware	Richard Kaapke
System Software	Alan Boyd
Assistant to the Publisher	Dan Yoder
Accounting Assistants	Mary Milam Lois Stickelmaier
Circulation	
Trial	Hal Schick
Subscriptions	Deirdre Booth Judy Weinstein
List Maintenance	Pat Adams Harry Richardson
Dealer Sales	Donna Siebert
Advertising	
West Coast Sales	Softalk Box 60 North Hollywood CA 91603 (213) 980-5074
East Coast Sales	Paul McGinnis Paul McGinnis Company 60 East 42nd Street New York, NY 10017 (212) 490-1021
Midwest and Rocky Mountain Sales	Ted Rickard John Sienkiewicz Market/Media Associates 435 Locust Road Wilmette, IL 60091 (312) 251-2541

Cover photo by Kurt A. Wahlner.

Composition by Photographics, Hollywood, California. Printing by Volkmuth Printers, Saint Cloud, Minnesota.

IBM and Personal Computer are trademarks of International Business Machines, Armonk, New York.

Softalk for the IBM Personal Computer. Volume 1, Number 4. Copyright © 1982 by Softalk Publishing Inc. All rights reserved. ISSN: 0733-2173. *Softalk for the IBM Personal Computer* is published monthly by Softalk Publishing Inc., 11021 Magnolia Boulevard, North Hollywood, CA 91601; telephone (213) 980-5074. Second-class postage pending at Saint Cloud, Minnesota, and additional mailing offices.

Postmaster: Send address changes to Softalk/IBM, Box 60, North Hollywood, CA 91603.

Subscriptions: Complimentary trial subscription to all owners of IBM Personal Computers in the USA. If you own a pc, but you aren't receiving *Softalk for the IBM Personal Computer*, send your serial number and mailing address to Softalk/IBM Circulation, Box 60, North Hollywood, CA 91603. Non-pc owner subscriptions: $24 per year. *Softalk for the IBM Personal Computer* is totally independent of International Business Machines. Back issues: $3.

Problems? If you haven't received your *Softalk for the IBM Personal Computer* by the fifteenth of the month, or if you have other problems with your subscription, Hal Schick can help out. Call (213) 980-5074.

Moving? Send new address and old to Softalk/IBM Circulation, Box 60, North Hollywood, CA 91603; telephone (213) 980-5074.

crosstalk

Questions for the Editors

Can you convince Roger Wagner to write an Assembly Lines column for the IBM? I'm finding the 8088 to be very trying.

Tim Born, Naperville, IL

We expect to be adding an assembly language column before long, although Wagner will probably not be the author.

In your June issue you mention in the summary of software sales on page 60 that the sixteenth largest seller is the *Dow Jones Portfolio Evaluator*. To the best of my knowledge, the portfolio evaluator package was designed by Apple for use only on its machines. I suspect that you are really talking about the *Dow Jones Reporter* package that is currently available through IBM.

Robert E. Klose, Waverly, NY

Looks like we did get the name wrong. Dow Jones Reporter *is correct.*

Questions for the Readers

According to my understanding, the Basic language will access only 64K bytes of memory at one time. I purchased my Personal Computer with 64K main board memory and an add-on 64K board. How can I make Basic access my entire 128K of memory?

D. W. Hornbrook, Jr., Marshall, AR

I am in the position of teaching people to use the pc. With a single monitor, only a few persons can see the screen. I'd like to be able to teach classes of ten to fifteen people. It appears that I have to purchase a color/graphics card and either a modulator (to which, presumably, a number of TV sets could be attached) or a single color monitor, with poor resolution. Surely there must be an easier way to tap the video to the IBM monitor and use it to drive one or more standard monitors. Any ideas?

L. T. Finch, Oak Brook, IL

My primary interest as a metallurgical engineer is the development of user-friendly process design and evaluation software for the mining industry. I would be grateful for any assistance you might provide in putting me in touch with users having similar interests. I would be more than happy to learn about any user groups that are at least oriented toward engineering applications development in general.

J. Mark Richardson, Union City, CA

I haven't been able to figure out how to get my Epson MX-100 to underline from *EasyWriter 1.0*. I've tried every interpretation of the manual instructions that I could think of.

That's not the only problem I'm having with printer commands. IPCO's June newsletter has extensive information from a member on Basic printer control commands. It would be a service if someone would put together a series of charts spelling out the printer control commands in Basic, *VisiCalc*, *EasyWriter*, etc., which work with different printers.

Henry F. Cooke, Bethesda, MD

How would I go about forming a user group in my area? Information on how to start a club would be appreciated.

F. A. Choudhri, Eau Claire, WI

Has anyone tried *Zork I* on a system with a black and white monitor and a color/graphics adapter? That's the system I'm using (the monitor is an Amdek 300G), and while it runs, the text is illegible. I couldn't find any way to tell it not to try to display color.

Jeff Spahn, Reseda, CA

How can I list a file when confronted with "Direct statement in file"? Is the LOF function a reliable way of determining the end of a random file, or is there a better way?

Michael Truffer, Deltona, FL

Review Advice

I hope *Softalk* will tell what is wrong with the software it reviews and not allow its rating scale to inflate to where excellent is a low score. If good reviews of software are your forte, then you can count on me to subscribe when the complimentary subscription expires.

Dan Barker, Yorktown, VA

Raves

I'm the data processing manager of an IBM 4341 shop and have fostered the use of the microcomputer in the company to alleviate some of the backlog of current projects, a defensive move. *Softalk* is an authentic and usable professional means of communicating information regarding the pc. Your articles are well researched, informative, and useful. I thought I was an expert on DOS until I read the recent article in your July 1982 issue.

J. A. Vilardo, Orland Park, II

My wife and I are both beginners with computers and we found the magazine very informative. Even after reading the Guide to Operations, we found operations in your Beginners' Corner that were not listed.

Robert P. Jernigan, Ewa Beach, HI ▲

NEW!

Kiss the 5¼" floppydisk goodbye!

Amdek has revolutionized data storage for personal computers with the new AMDISK-3 Micro-Floppydisk drive system. The system consists of 2 drives and a power supply, fully compatible with 5¼" floppy disk drives. The 3" disk is encased in hard plastic, protected from dust and fingerprints, and it's easy to mail.

Just write, or call to receive our data sheet on the new AMDISK-3 Micro-Floppydisk Cartridge system.

- 1 Megabyte (unformatted) storage capacity.
- Track-to-track compatible with 5¼" floppy-disk drives.
- 3" shirt-pocket sized disk cartridge.
- Drive has built-in power supply.

2201 Lively Blvd. • Elk Grove Village, IL 60007
(312) 364-1180 TLX: 25-4786

AMDEK CORP.

Amdek . . . your guide to innovative computing!

beginners' corner

by Craig Stinson

We left you hanging last month. You had just finished loading the *Samples* program from your DOS disk into memory when we were forced to adjourn. You had also typed *list* and summoned to your screen a display of the Basic commands, or statements, that make up the *Samples* program. The next logical step, of course, would have been to type *run*, hit enter, and watch *Samples* go into action on your screen.

Load, *list*, and *run* are all Basic commands that are most commonly used in immediate execution mode. That is, these commands generally are used outside of a Basic program. You could also use them from within a Basic program—you could, for example, write a program that would list itself—but their more typical use is from without.

We're not going to get into programming per se in this column; there are hundreds of books that do an excellent job of covering that subject. We will, however, have more to say about the use of elementary program-handling commands like *load* and *run*. Before doing that, however, let's take one more wide-angle view of what was going on in your computer when we left off last month.

You had booted DOS. When you got the A> prompt from DOS, you typed *BasicA* and hit enter. That loaded Advanced Basic into memory. Then, facing Basic's Ok prompt, you loaded the *Samples* program into memory. Let's assume for the sake of completeness that you also had typed *run*.

At this point you would have had an application program, *Samples*, running within the context of a programming language, Basic. The language, itself a program, would have been running within the context of an operating system, called DOS. And this whole constellation of software—DOS-Basic-*Samples*—would have been operating within the framework provided by the hardware of the IBM Personal Computer.

All these functionally distinct software elements are designed to work with each other. A program like *Samples* that's written in Basic must have some form of Basic interpreter present in memory in order for it to run (next month's column will have more about what a Basic interpreter is and about the different kinds of Basic available to you on your pc).

Advanced Basic, meanwhile, requires the presence in memory of DOS. There are, as you may know, a zillion versions of the Basic programming language. Aside from certain small-scale versions that lack the capacity to deal with disk files, they all work in concert with some particular operating system. The two Basics on your DOS disk—Basic and BasicA—require DOS. There are other Basics that function with such operating systems as CP/M-86 and the UCSD p-System.

And finally, DOS itself is tailored for the design of the IBM Personal Computer. The nomenclature could be a little confusing here. The name DOS is not unique; it stands for Disk Operating System, which, of course, is a pretty generic term. So there's a DOS that runs on the Apple II, a DOS that runs on the TRS-80, and so on. They are not all the same animal, though they serve the same general functions for their respective machines.

DOS in the context of your computer is short for PC-DOS, which is one version of an operating system called MS-DOS (the MS stands for Microsoft, the company that owns the rights to the system). MS-DOS runs on a number of machines that, like the IBM Personal Computer, are designed around the 8088 microprocessor chip. But PC-DOS is tailored for the hardware design of the Personal Computer; aside from the pc, it probably only works on a few machines that were designed to mimic, or emulate, the pc.

So you have these several elements of software working in harmony with one another when you're running a program like *Samples*. The part you see is the application program; the programming language, if present (many programs are not written in Basic and don't require its mediation), is lurking in the background. And farther in the background is the operating system.

Transparent Computation. The parts you don't see are sometimes said to be "transparent to the user." Such jargon is a little misleading; a factor operating in the background is not transparent, but invisible. Nevertheless, transparent is the expression you'll encounter. All it means is that something useful is happening and you don't have to worry about it.

Part of what's going on—transparently, as it were—is that the various functional elements are interacting with each other. For example, when you're looking at the Ok prompt and you type the command *load"samples*, you are issuing a command in Basic, but Basic has to call upon the services of the operating system to execute your command. That's because the operating system is in charge of relations with the disk. Similarly, if you use a program like Micro Lab's *Tax Manager* that's written in Basic and that creates and uses data files (files that contain information, rather than instructions), you'll be invoking DOS every time you move information between memory and your disk system.

Your application program, Basic (if present), DOS, and any data you're working with all share quarters in memory. The microprocessor looking at these apparently distinct elements sees only one large program. In fact, what it sees is a large stream of numbers.

Everything stored in memory is encoded in a form of numbering called binary—a subject to which we'll return. Each address in memory is like a little mailbox that can hold one message—a value

1st Class Mail™ It does everything but lick the stamp.

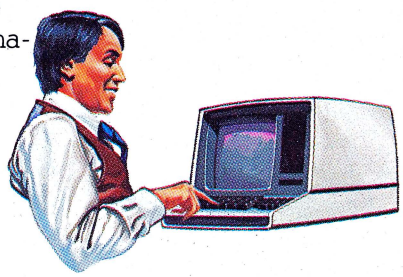
We've made a mailing list program worthy of today's remarkable microcomputers. It's called *1st Class Mail*™—but you can use it for anything you'd keep on a card file, like your car's servicing history or your address book. And it's not only the most flexible mail list/utility data base you can buy—it's also incredibly easy to use.

You create the label.

Since you're not stuck with the same old name/company/street/etc. format, you can make the label look like whatever you want. And with the new feature that lets you print right on your envelopes, you not only save time—you give your correspondence a very professional look.

Super search and sort.

1st Class Mail™ lets you search and sort in any of 12 different categories. So you can organize your mailings by zip code (up to 10 digits), alphabetical order, state, company, any way you want. Fast. And if you're using *1st Class Mail*™ for another type of data base, like a record collection, you can sort by whatever categories you choose— from String Quartets to Rhythym and Blues. It's extremely flexible.

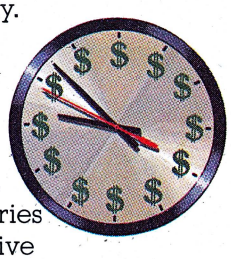


Personalized entries.

One of the ways we've made *1st Class Mail*™ especially "user friendly" is by letting you create your own headings for each entry line. So your computer speaks *your* language.

The form letter feature.

1st Class Mail™ is compatible with many word processors, such as Screenwriter II,™ Applewriter II,™ Applewriter III,™ Pie Writer,™ and WordStar/Mailmerge.™ Just a few simple commands and you can combine your form letters with your mailing list for professionally formatted documents. All automatically.

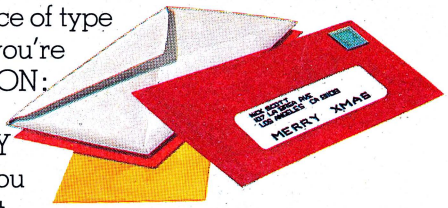


Automatic entry.

If time is money, this feature translates minutes directly into dollars. It automatically includes, or searches for, particular categories in each entry you record. Repetitive typing and sorting becomes a push-button joy.

The special message line.

Now you can put a special message anywhere on the label—in a choice of type styles. Whether you're saying ATTENTION: SALES MANAGER or MERRY CHRISTMAS! you can do it automatically. And right on the envelope.



Stop by your Continental Software dealer. Or call us today to find out all about *1st Class Mail*.™ No matter what you're cataloging, from a household inventory for insurance purposes to a 25,000 person mailing list, *1st Class Mail*™ is the best program you can buy.

It's true, you still have to lick the stamp.

But we're working on it.

Available for: Apple II,™ Apple III™/III with Profile,™ IBM-PC™/IBM-PC with Tecmar™ hard disk/IBM-PC with Davong™ hard disk.

Apple II/II+, Applewriter II, Applewriter III are registered trademarks of Apple Computer, Inc. Screenwriter II is a registered trademark of On-Line Systems. Pie Writer is a registered trademark of Hayden Book Co. WordStar Mailmerge is a registered trademark of MicroPro, Inc. Tecmar is a registered trademark of Tecmar, Inc. Davong is a registered trademark of Davong Systems, Inc. IBM-PC is a registered trademark of IBM Corp.

Continental Software

Continental Software, 11223 South Hindry Avenue, Los Angeles, California 90045, Telephone (213) 417-8031

new faces

15 type styles for the IBM PC and Apple II

With this program your dot matrix printer can output 15 large typefaces.

Type Faces generates distinctive presentations, fancy lettering, invitations, easy to read output, and over 100 symbols. Reduced printout on a copier gives you letter quality text... an inexpensive typesetter.

Type Faces comes with its own simple text editor and is compatible with most word processors.

Actual dot matrix printout	Reduced dot matrix printout gives you letter quality text
ABC abc	ABC abc
ABC abc	ABC abc
ABC abc	*ABC abc*
ABC abc	ABC abc
ABC abc	ABC abc

Alpha software products available at participating **ComputerLands**
Or call us for your nearest dealer: 617 229-2924

alpha SOFTWARE CORP

6 New England Executive Park • Burlington, Massachusetts 01803

Type Faces supports Epson and IDS dot matrix printers. Type Faces: $95

between 0 and 255. The microprocessor interprets some of these values as instructions and others as data objects upon which to carry out instructions. The distinction between instructions and data is by context only. In one setting a given number may tell the 8088 to take some action and in another it may only represent an operand—a piece of data upon which the 8088 is to act. In another setting a number may only tell the processor to move on and read another memory location.

The computer, scanning through its memory, is like a kid on a scavenger hunt. Ordinarily, when it's finished executing an instruction, it moves on to the next highest address, and so on. But now and then it finds an instruction to jump off to some distant part of its domain, either to get a piece of data there and return or to begin following instructions from that point.

Once in a great while something may go wrong; a program may have some faulty logic in it and as a result the computer may get stuck reading some particular set of addresses over and over again. This condition is called an infinite loop. The symptoms are a general lack of responsiveness on the part of your computer. In other words, it won't hear a word you say. How could it? It's busy running around in circles.

Unfortunately, even the reset sequence (hitting control, alt, and delete at the same time) won't revive the Personal Computer from certain kinds of loops. That's because the pc has what's called a software reset; the reset sequence doesn't affect the microprocessor directly the way it does on some other computers that have hardware resets. When your pc is frozen, it won't even see the reset, so there's no recourse but to turn the thing off and start over. What's bad about that, of course, is that if you have any data in memory that you haven't sent out to disk, that data will be gone the moment you turn the power off.

Garbage Collection. And speaking of data, there's another phenomenon concerning memory that you need to be aware of. It's called garbage collection.

If you have a machine with 64K or more of memory, you can usually have DOS and Basic or BasicA loaded and still have enough memory addresses for a fairly large application program. But programs usually create and use data. For example, if you run IBM's *Mortgage* program—one of the demonstration programs on the DOS disk—you'll be asked to enter some principal amount, an interest rate, a loan period, and so on. The program will calculate and display a lot of numbers on your screen. Those numbers sit somewhere in memory.

If you're working with, say, an accounting program and you enter a lot of information into memory, you may eventually run out of room. In that situation the computer may take a quick look through its memory and throw out any data it no longer needs to hold. Your computer will appear to be dead to the world while this is going on, but the process only takes a second or two at the most and it all happens, as they say, transparently.

Most accounting programs save data to disk periodically while you're working, so they're able to reclaim substantial portions of memory when things get full. This is not the case with some other kinds of programs. Most word processors, for example, retain in memory the entire document you're creating or editing; they only ship your work off to disk when you say to. With such programs you may indeed run out of memory. A well-written program, however, will give you some warning before that happens.

Enough about memory. Let's get back to Basic and look some more at those program-handling commands. Whether or not you decide to take up programming yourself, you'll need some familiarity with these commands in order to be comfortable using other people's programs. You've already encountered *load*, *list*, and *run*. We'll review those and also comment on *save*, *files*, *name*, and *kill*.

Load. *Load*'s function is to bring a Basic program into memory. The first thing *load* does is check to see that the program you wish to load actually exists. If it finds the program you request, it then deletes from memory any Basic program that may have been there at the time you issued the *load* command and brings the new program into memory. If the program you had been running immediately before issuing the *load* command happened to have used data files, those data files are also deleted by *load*.

A word or two about file types is in order here. Anything that you put on disk gets stored there in the form of a file. A file is just a collection of information unified by a name. The information may be stored on disk in one contiguous block, or it may be in several separate pieces. However it's arranged physically on disk, to the computer it's one logical unit. The operating system keeps a directory on disk, called the file allocation table, that enables it to find its way to the appropriate disk sectors when you request a load or save.

Files break down by content into two large categories: program files and data files. Program files contain executable instructions, and data files contain information used by programs.

For example, if you do accounting on your computer, you presumably load some kind of program file which, in turn, creates data files. How many and what kind it creates is up to the program. It might perhaps make one file containing your budget and another containing transaction information. Or it might create a number of separate transaction files, one for each month.

In any case, data files make information available to programs but don't contain any instructions that can be executed directly by the computer. The Basic *load* command is for Basic program files only; if you try to load a data file, you'll get an error message that says, "Direct statement in file." You'll get the same error message if you attempt to load a program file that isn't coded in Basic.

Which raises a question—how do you know from a file's name whether it's a Basic program file, some other kind of program file, or a data file? Usually by the file's suffix, or extension. It's customary for the names of Basic program files to end in .bas. The period and three letters following are called an extension. When a Basic program file is first created and put on disk, if no extension is appended to the file's name, Basic will give it the extension .bas automatically. So the only way a Basic program file might not have the .bas tag would be if someone came along later and renamed the file to omit the extension, or if some extension other than .bas had been given to the file name when the file was created in the first place.

For all intents and purposes you can assume that a Basic program file will end in .bas. When you load the file from disk, you don't need to include the extension when you type the name of the file. On the other hand, you may include it if you wish. If you happen to have a Basic program file on disk that has some other extension, you'll have to specify its name in full.

The complete form of the command to load a Basic program, then, is the word *load*, followed by a double quote mark, followed by the name of the file, with or without the extension .bas. Ending quotes are not necessary, except under one condition.

Load ordinarily brings a program into memory where you can look at it, change it, send it back to disk, or run it. It's possible with the *load* command, however, to bring a file into memory and run it directly. You do that by tacking on closing quotes after the file's name, followed by a comma and the letter *r*. This procedure accomplishes the same thing as the *run* command, which we'll come to momentarily, with one difference: it leaves intact any data files that happen to be in memory. The *,r* option for the *load* command allows you to bring a program into memory and make available to it data files that had been used by the program previously run.

One more point needs to be made about *load*. When you're in DOS, before you bring up Basic or Advanced Basic, you get a prompt from the operating system that consists of a capital letter followed by the greater-than symbol. The letter names the currently logged disk drive—the one that DOS considers active. Any commands you issue that involve the disk will be carried out on the

logged drive, unless you request otherwise.

Whatever drive was the logged drive when you typed *Basic* or *BasicA* is still the active drive after Basic takes its place in memory. So a command such as *load* will normally be carried out on that logged drive. If the logged drive is A and you want to load a program residing on a disk in drive B, you have to add a *b:* after the quote marks and before the file name. The *b:* is called the drive specifier; it mustn't have any spaces within it or around it.

The same remarks apply to the other Basic commands that involve the disk; the commands will be directed toward the logged drive ordinarily, but you can change that by including a drive specifier before the file name.

Run. The *run* command does one thing if you type it by itself and another if you put a program name with it. By itself, *run* causes the computer to start executing whatever Basic program is in memory. Execution will begin at the lowest-numbered program line. It is possible to start a program running at some point other than the lowest-numbered line; you do that by following the word *run* with the line number where you want to start. *Run 40*, for example, will get a program running from line 40 on.

Run followed by a file name brings a program in from disk and starts it running at the lowest line number. Like *load*, *run* used in this fashion erases from memory any program that may be sitting there when the command is issued. It also erases any data files currently in memory. As with *load*, there's a *,r* option available for the *run* command that brings in a program and runs it but also keeps any data files in memory available to the incoming program.

The syntax for the *run* command when it's used to get a program from disk is the word *run*, followed by double quotes, followed by the name of the program. If the program ends in .bas, as most Basic programs do, you may omit the extension or not, as you choose; if the program is stationed in some drive other than the logged drive, you need to include a drive specifier. To use the *,r* option, follow the program name with another double quote, a comma, and the letter *r*.

Save. *Save* does the opposite of *load*: it sends a program from memory to disk. Actually it copies the program to disk; the program remains in memory after the save has been done. The syntax is similar to the syntax for *load*: type *save*, double quotes, a drive specifier if you want to send the program to a drive that's not the logged drive, and the program name. If you don't put an extension on the program name, Basic will automatically tag it .bas.

There are two options for the *save* command. Adding *,a* after a closing double quote mark saves the program on disk in a file format that takes up more disk space than the normal format but allows the program to be used as a data file by other programs. Putting a *,p* after the closing quote saves the file in a form that can't subsequently be listed. This is one way of frustrating snoops and pirates. If you try to list a program that's been saved with *,p*, you'll get the message: "Illegal function call."

The *list* command displays on screen the Basic statements making up the program currently in memory. Hitting control-num lock stops a long list from scrolling, so that you can get a look at it. Hitting any other key makes the listing resume. Hitting control-scroll lock terminates a list.

If you only want to see part of a program list, you can specify the range of line numbers desired after the word *list*. *List 300–600*, for example, will display lines 300 to 600 inclusive. *List 300–* will show everything from line 300 to the end of the program, and *list –300* will show everything up to line 300.

Other Commands. The last three commands we'll look at, *files*, *name*, and *kill*, are functional analogs of commands that also exist in DOS.

Files, like DOS's *dir*, displays on screen the contents of a disk. You'll notice some difference between the two commands; *files* doesn't provide quite as much information as *dir*, but it puts more file names on the screen at one time. If you only want to check to see if a particular file is on disk, you can follow the *files* command with double quotes and the name of the file you're hunting for.

To catalog a disk that isn't logged, you need to type *files*, a double quote, an asterisk, a period, and another asterisk, as in: *files"b:*.**. The asterisks are what are called wild card characters; in this context they have the effect of your saying, "Show me what you got on drive b." The wild card characters are handy for eliciting selective directory lists and for some other purposes, too. Since they're used by DOS as well as by Basic, you'll find a full discussion of them in the DOS manual, pages 2-9 through 2-11.

Basic's *name* command parallels DOS's *rename*. Both commands change the name of a file stored on disk. The file's physical disk locations and contents are unaltered by the process. The syntax is as follows: *name*, followed by double quotes, followed by the current name of the file, followed by another double quote, the word *as*, yet another double quote, and the new name of the file. If the file is located on a nonlogged drive, the drive specifier must be in there as well, right after the first quotes. If the file you're renaming is a Basic program whose present name ends in .bas, the extension must be included in this command.

Finally, Basic's *kill* is the equivalent of DOS's *erase*; it removes a file from disk. Actually it only removes an entry from the disk directory; the contents of the file are not physically erased, but the space they occupy is made available for new files. The syntax is *kill*, double quotes, name of file; the .bas, if present, must be included in the command.

That takes care of the most frequently used Basic file-handling commands. Next month we'll talk more about what Basic is and how it compares to other programming languages. We'll also discuss the various versions of Basic available on your Personal Computer. ▲

NF SYSTEMS

★Master your PC with our new design aids:

FILE MASTER A comprehensive disk file maintenance program using full screen displays. Creates reports by any field. Supports VICICALC® files! **79.95**

TEXT MASTER A document formatter which offers 40 commands to customize text. Includes indexing and table of contents capabilities! **99.95**

SCREEN MASTER A fantastic utility for BASIC programmers – defines full screen displays for input/output, supports all keys, colors, and allows field protection. **79.95**

FILE COMPARE A file comparison utility using multiple parameters for compare, and sending display to screen or printer. **34.95**

SORT 80 An assembler sort subroutine, callable by a BASIC program. Variable length fields. FAST! **34.95**

DISKETTE CATALOG Create a master catalog of all your diskettes! If PC - catalog is created from disk. For other diskettes, enter from keyboard. May change and/or search for a particular file. **A must!** **34.95**

★Master your household with:

HOUSEHOLD AIDS Includes Household Inventory, Personal Statement, VCR Index, Resume, Name/Address/Phone. **49.95**

CHECK REGISTER Up to 40 ledger/budget headings. Print Check Register or Totals by ledger/by month. **39.95**

STOCK PORTFOLIO Keep all transactions by company name and date. Print entire portfolio or transactions by month or year. **39.95**

NF SYSTEMS, LTD.
P.O. Box 76363
Atlanta, GA 30358

VISA MasterCard

Catalog Available

IBM Personal Computer

To order, call collect 9-5 404-252-3302

SIMPLICITY:

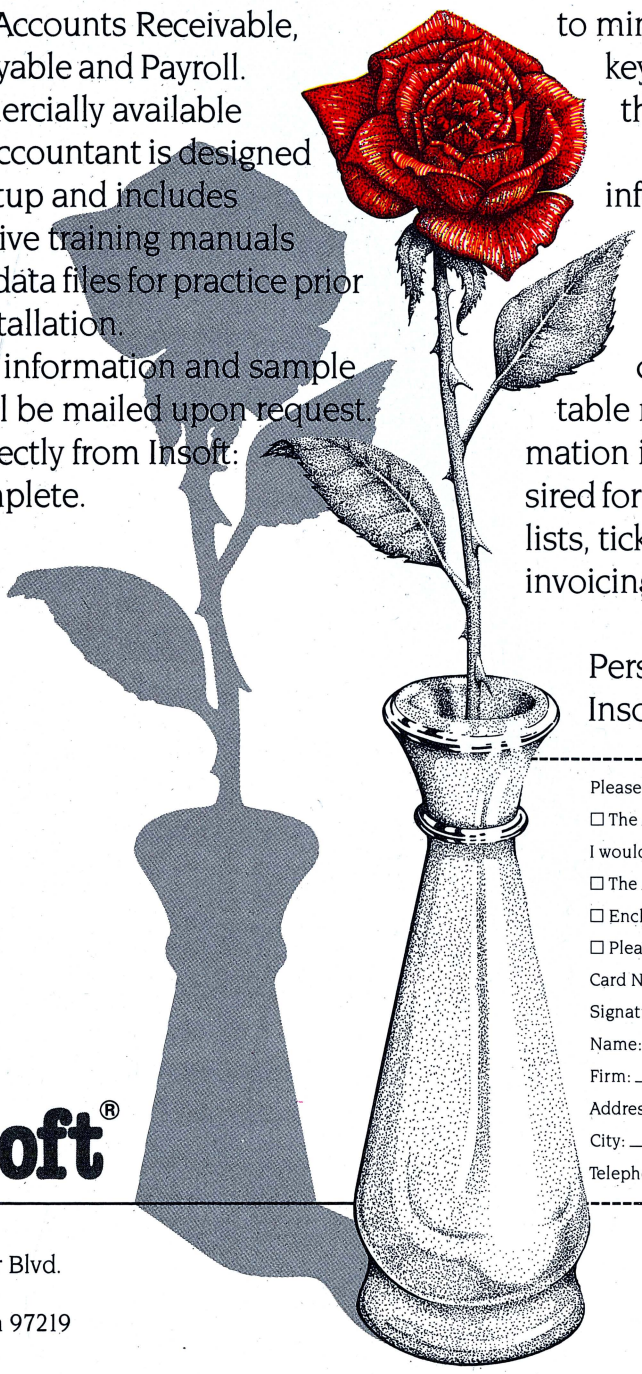
THE ULTIMATE SOFTWARE SOPHISTICATION

THE ACCOUNTANT™ is your alternative to costly and inflexible accounting systems. Organized into four packages, The Accountant is fully menu driven and includes General Ledger, Accounts Receivable, Accounts Payable and Payroll. Using commercially available forms, The Accountant is designed for easy startup and includes comprehensive training manuals and sample data files for practice prior to actual installation.

Detailed information and sample printouts will be mailed upon request. Available directly from Insoft: $365.00, complete.

DATA DESIGN™ is, quite simply, the most easy-to-use data base manager available. Fully menu driven and complete with help screens, Data Design uses the function keys to minimize learning time and operator keystrokes. Designed specifically for the Personal Computer, Data Design is as powerful as it is simple. All information is entered, viewed, changed and printed through user-created forms. Each form may have up to 40 actual and computed fields, and since each table may use as many as 26 forms, information is always ready in precisely the desired format. Data Design is ideal for mailing lists, tickler files, customer/patient records, invoicing and accounts receivable.

Available in October at your local Personal Computer dealer, or from Insoft: $225.00.



Please send me more information about:

☐ The Accountant ☐ Data Design

I would like to purchase:

☐ The Accountant for $365.00 ☐ Data Design for $225.00

☐ Enclosed is my company check/money order.

☐ Please charge my ☐ VISA ☐ MasterCard

Card No.:_____ Exp. Date:_____

Signature:_____

Name:_____

Firm:_____ Title:_____

Address:_____

City:_____ State:____ Zip:_____

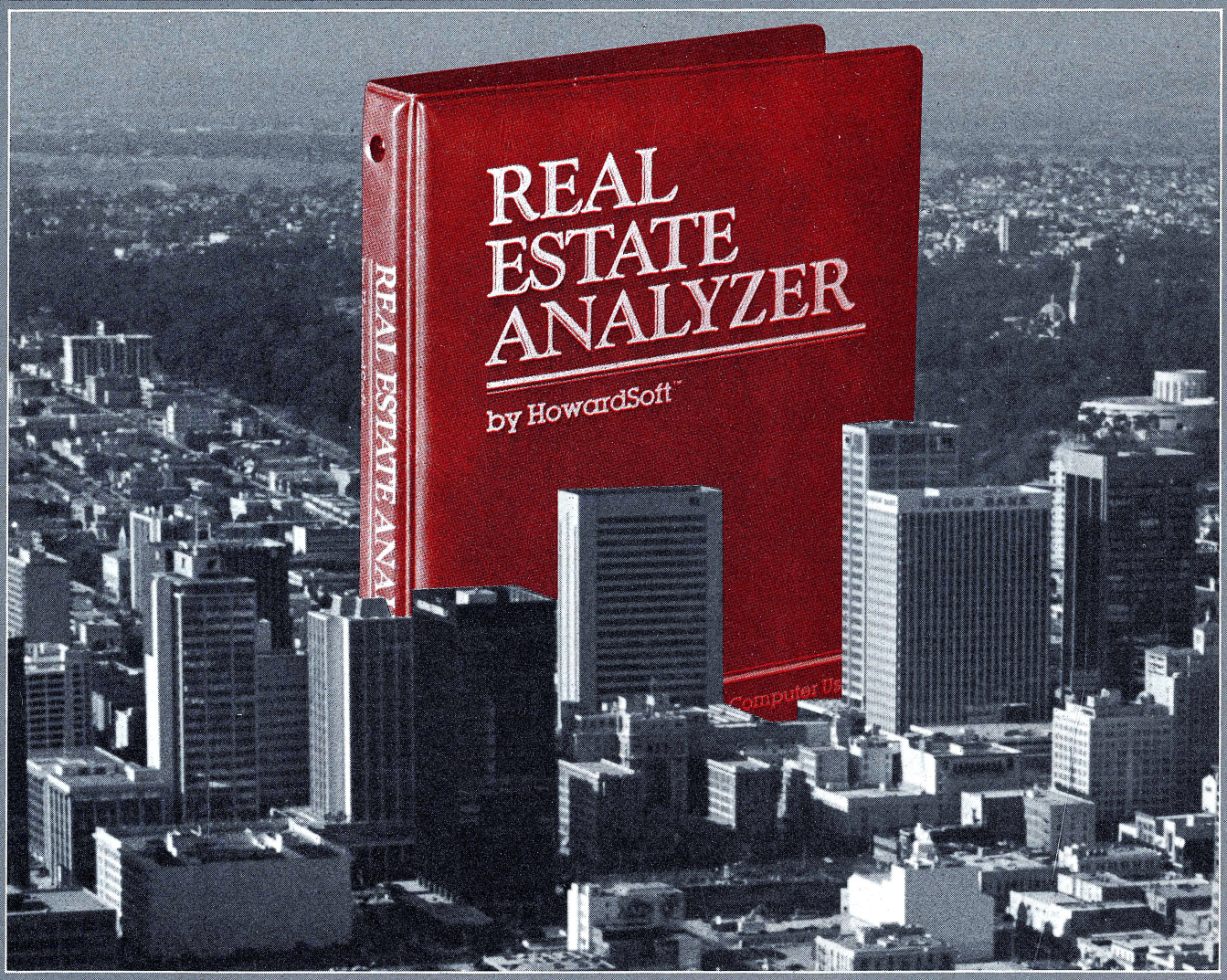
Telephone:_____

insoft®

Insoft, Inc.
10175 S.W. Barbur Blvd.
Suite 202B
Portland, Oregon 97219
503/244-4181

Both packages support the IBM (Epson) printer and a variety of hard disks.

THE BEST STANDS OUT



*PRACTICAL DECISION-MAKING
aids buy/hold/sell decisions by projecting cash flow & profitability; computes ROI, IRR, ROE, and FMRR.

*REALISTIC SIMULATIONS
models economy based on 5 separate components of inflation for realistic "what-if" studies

*UP-TO-DATE WITH CURRENT MARKET
takes into account rent control, runaway inflation, money rates, creative financing, and complex lease packages

*UP-TO-DATE WITH TAX LAWS
includes latest tax laws for depreciation (including ACRS), recapture, & capital gains

*PROFESSIONAL PRINTOUTS
detailed 10-year tables of cash flow and ROI in personalized, paginated report format

*ERROR TOLERANT
designed for ease of use by newcomers to computers; fully tested by real estate professionals

*INSTRUCTIVE MANUAL
100-page book includes easy-to-follow examples and detailed instruction in investment analysis

The best investment clearly stands out when you use computer-aided decision-making with the Real Estate Analyzer by HowardSoft. Applying the latest techniques in investment analysis, this software produces detailed, objective projections of cash flow and overall investment return for any commercial or residential property. Even the most dissimilar deals are compared fairly because inflation rates, creative loan packages, complex depreciation schedules, involved leases, and complex tax laws are all taken into account.

Now, better than ever
$195 for Apple
(48K and Applesoft)
$250 for IBM PC
(with 64K and IBM DOS)

Ask about the HowardSoft Tax Preparer and our other financial software.

HSS Howard Software Services
for the SERIOUS Personal Computer User
8008 Girard Ave., Suite 310 | La Jolla, CA 92037 | (714) 454-0121

THE PROFIT PLOT

by Shirley Fisher

It appears that inflation is here to stay, at least for the next twenty-five to thirty-five years. The income that meets people's present needs won't have the same buying power in the future. This month we'll use an electronic spreadsheet to help us determine the monthly income that will ensure an adequate standard of living later on.

Ideas of what constitutes an adequate income vary. So do people's abilities to set money aside each month for the attainment of future financial goals, the number of years they plan to work before retiring, and their beliefs about what constitutes a sound investment strategy.

Preparing for the future means exploring some key issues now. Questions worth asking yourself include:

When do you plan to retire? Given the current rate of inflation, what income will adequately meet your retirement needs? How much money can you manage to set aside now on a monthly basis? Which of the available investment options will give you the best return on your money?

Social Security Isn't Enough. Lots of us are not very realistic about the amount of money we'll need each month in order to enjoy an adequate standard of living in retirement. Many people who have worked hard all their lives now receive only a monthly Social Security check of $600 or less. If this is their sole source of income, they are already considered to be living at or below the poverty level.

But the Social Security program was never intended to be the sole source of retirement income. It was partly as an attempt to help people close the gap between existing social aid programs and actual projected needs that company retirement plans came into being.

Some people believe that Social Security payments and their company's retirement fund (if they are fortunate enough to have one) will be enough. The annual report they receive says that they will get a large amount of money each month when they retire in, let's say, the year 2001, and this reassures them that all is well. True, in light of today's purchasing power, the amount looks quite large enough, but the present rate of inflation guarantees the amount won't seem so great in the year 2000. Even with both Social Security and a company retirement plan, retirement income won't be adequate in most instances.

Inflation History. So how do we go about figuring out how much money we'll need in retirement? Considering some facts and statistics from the past can be an instructive and worthwhile first step.

Some of you may recall a time when five cents or seventeen cents would buy a loaf of bread. In recent years, thirty-five cents bought a loaf of bread. Now, in the eighties, a loaf of bread costs nearly a dollar! This same type of price increase has happened in most other consumer goods as well. And although it's possible to reduce your expenses in certain areas as you grow older, chances are you can look forward to paying the going rate for such vital items as food and clothing.

A detailed look at what's happened to the American dollar over the last thirty years or so will give you a clearer understanding of what's probably going to be true in the next thirty. Statistics from the Department of Labor Consumer Price Index reveal that in 1956 the dollar had purchasing power of only ninety-six cents compared to its purchasing power five years before. By 1961, a dollar could buy only eighty-eight cents' worth of goods; in 1966, it could buy seventy-nine cents' worth; and by 1971, it could buy only sixty-three cents' worth of goods. In 1978, a dollar could buy you what forty cents would have purchased in 1951.

Get MORE out of VISICALC With

V-UTILITY $99.95

COPYRIGHT © 1981 YUCAIPA SOFTWARE Version 5.0

SPECIFY COMPUTER MODEL

NOW AVAILABLE FOR IBM PERSONAL COMPUTER (PC DOS)
APPLE II (DOS 3.2 and DOS 3.3)
TRS-80 MOD I, II, III and 16 (TRSDOS)

V-UTILITY CONSISTS OF ALL THESE VISICALC "USER FRIENDLY" UTILITY PROGRAMS ON ONE DISK RUN BY INDEX AND PROMPTS.

V-PRINT SELECT FROM 1 TO 16 COLUMNS AND PRINT THEM IN ANY ORDER. YOU MAY SPECIFY THE PRINT WIDTH OF EACH OF THE COLUMNS INDIVIDUALLY AND ALLOCATE FROM 0 TO 6 SPACES BETWEEN THE COLUMNS. A FANTASTIC UTILITY PROGRAM.

V-STAT PROGRAM COLLECTS DATA AUTOMATICALLY FROM THE VISICALC COLUMN AND CALCULATES NUMERICAL DISTRIBUTION, CORRELATION COEFFICIENT, REGRESSION ANALYSIS, CHI² TEST, AND T-TEST. YOU MAY SELECT THE COLUMNS FOR DATA ENTRY AND SPECIFY THE ROW# TO START AND ROW TO END DATA COLLECTION. PROGRAM IS EASY TO RUN.

V-PLOT THIS PROGRAM AUTOMATICALLY INPUTS DATA FROM A VISICALC COLUMN, PERFORMS AUTO SCALING THEN PLOTS EITHER 1 OR 2 COLUMNS ON A REGULAR LINE PRINTER (GRAPHICS NOT REQUIRED). IDEAL FOR ANALYZING UP TO 250 NUMERICAL DATA POINTS IN RELATION TO TIME.

V-OVERLAY PROVIDES THE FOLLOWING SELECTION OF OVERLAYS TO LOAD ON TO THE VISICALC SHEET. MOVING AVERAGES, EXPOTENTIAL SMOOTHING EQUATIONS, TIME SERIES TREND ANALYSIS, DATE COLUMNS.

YUCAIPA SOFTWARE
12343 12TH ST · YUCAIPA·CA·92399
PHONE (714) 797-6331 SHIPPING & COD $3.00

IBM, APPLE, TRS-80, and VISICALC are trademarks respectively of International Business Machines Corp, Apple Computer Inc, Tandy Corp, and VisiCorp

	Late sixties	Late seventies	Late eighties	Late nineties	2010
Groceries	$115	$200	$350	$600	$1,000
Mortgage payments	$150	$430	$1,200	$3,400	$9,500
Medical care	$100	$200	$460	$980	$2,100
Fuel	$125	$275	$615	$1,375	$3,075
New car	$3,200	$6,500	$13,000	$26,500	$54,000
House	$23,000	$52,000	$120,000	$277,000	$639,000

Table 1. Average costs of selected items.

Quite a drop in purchasing power over twenty-seven years, isn't it? And yet it's relatively mild when compared to what experts are predicting for the future, and it's certainly mild compared to the inflation rates other nations have experienced over the same time interval.

Table 1 shows the impact of inflation on the prices of certain specific items. Bear in mind that the figures shown are average monthly expenses for the so-called typical American family. And the projections for the next eighteen years assume only a moderate inflation rate of 7 or 8 percent. Double-digit rates would of course take those prices much higher.

Observe also that the figures quoted for new cars are not for super-expensive luxury models but for ordinary medium-priced cars with average equipment. And Californians, take note of the fact that the housing figures projected for the late eighties have already been surpassed on the West Coast, and we're still a half decade away.

Bracket Creep. It's tempting to think that wage increases will keep pace with inflation, so there really isn't anything to worry about. Maybe wages will grow as fast as prices and maybe they won't, but in either case, as your annual income increases you'll find yourself automatically pushed up into higher income tax brackets. You've probably heard this phenomenon referred to as "bracket creep." How does it happen?

It works like this. If you've kept the Form 1040 instruction booklets for the past few years, you can dig them out and see for yourself. Let's use $15,000 as an example of taxable income for the year 1975, and let's also assume an annual increase in taxable income of 10 percent. Using the Married Joint schedule, we can observe the following:

Year	Taxable Income	Total Tax	Percent of Taxable Income	Percent over $ Amount
1975	$15,000	$3,010	20	25/$12,000
1976	$16,500	$3,393	20.5	not given
1977	$18,150	$2,997	16.5	25/$15,200
1978	$19,965	$4,025	20	28/$19,200
1979	$21,961	$3,766	17	28/$20,200
1980	$24,157	$4,381	18	28/$20,200
1981	$26,573	$6,478	24	32/$24,600

As you can see, a person who made $15,000 in 1975 and whose income has grown by 10 percent each year has had to double his annual tax outlay in seven years. The percent of taxable income going to pay income taxes increased only 4 percent between 1975 and 1981, but what happened to the purchasing power of your dollar during that time? Did it buy 4 percent more in 1981 than it did in 1975? You bet it didn't! It bought less.

If we assume an 8 percent rate of inflation, you would need $23,790 now to give you the purchasing power of the $15,000 you had in 1975. If you increased your income each year by 10 percent, you would be ahead of the game, but just a little bit. Or would you?

In 1981, with income of $26,573, you needed $23,790 to purchase what $15,000 purchased in 1975, leaving you an increase in before-tax dollars of $2,783. But in 1975 taxes due were only $3,010. In 1981 taxes due were $6,478, a difference of $3,468. That means you would actually have had $685 less with which to purchase the same goods and services you bought in 1975.

Income at Retirement. Once you're retired with little or no taxable income, you aren't going to be concerned about "bracket creep." Your big concern will be the shrinking power of your retirement dollar. How much money will you need when you retire? Let's see how we can figure that out.

If the inflation rate is 8 percent, then for each year in the future your dollars will have to grow by 8 percent to maintain their present purchasing power. Using $1,000 as an example sum, in one year you would need 1.08 times $1,000, or $1,080. The next year you'd need 1.08 times that amount, or $1,166, and so on.

So how much money would you need if you were to retire today? In five years, your original $1,000 would require $1,469 to retain its purchasing power. Suppose you want to have $20,000 worth of purchasing power five years from now. Divide $1,469 by $1,000 to get the conversion factor, then multiply that result, 1.469, by $20,000. Did you get $29,380?

If your retirement target is ten years away, your conversion factor would be 2.158 and your need would be $43,160. In fifteen years the factor would be 3.172 and your need would be $63,440. To maintain the purchasing power of $20,000 twenty years hence, you'd need to have $93,000.

We can set up an electronic spreadsheet to give us a clearer picture of the amount of money needed for various lengths of time, based on different inflation rates and today's needs. A handful of formulas will provide us the information we need. We can let the inflation rate, years to retirement, and today's needs vary as user input; all other values will be calculated from these three variables.

Figure 1 shows the labels and calculated results. The formulas and formats are as follows:

International Computers

Software--IBM PC/ Apple II & III

Go-Forth--Know Forth in two hours? Well, maybe not, but certainly get acquainted with the help of Computer Assisted Instruction. Forth language & editor included. **$59.95**

* **Speedup-Plus**--Faster than Speedup--Virtual memory for the IBM. **$199.95**
* **79-Standard FastForth**--FastForth® is faster than any microcomputer Forth. **$499.95.**
* **79-Standard FullForth**--Imagine 512K of Forth application code. Not available anywhere else! **Available soon**
* **SerPrinSoft**--Does your Serial Printer run at full Baud Rate? If not, you need SerPrinSoft. Hardware & XON-XOFF protocol supported. **$79.95**

Mister Lister--"Space Shuttle of Mailing List Programs"--Sort over 2,000 names in one minute! Fully compatible with WordStar‡ & MailMerge‡. **$79.95**

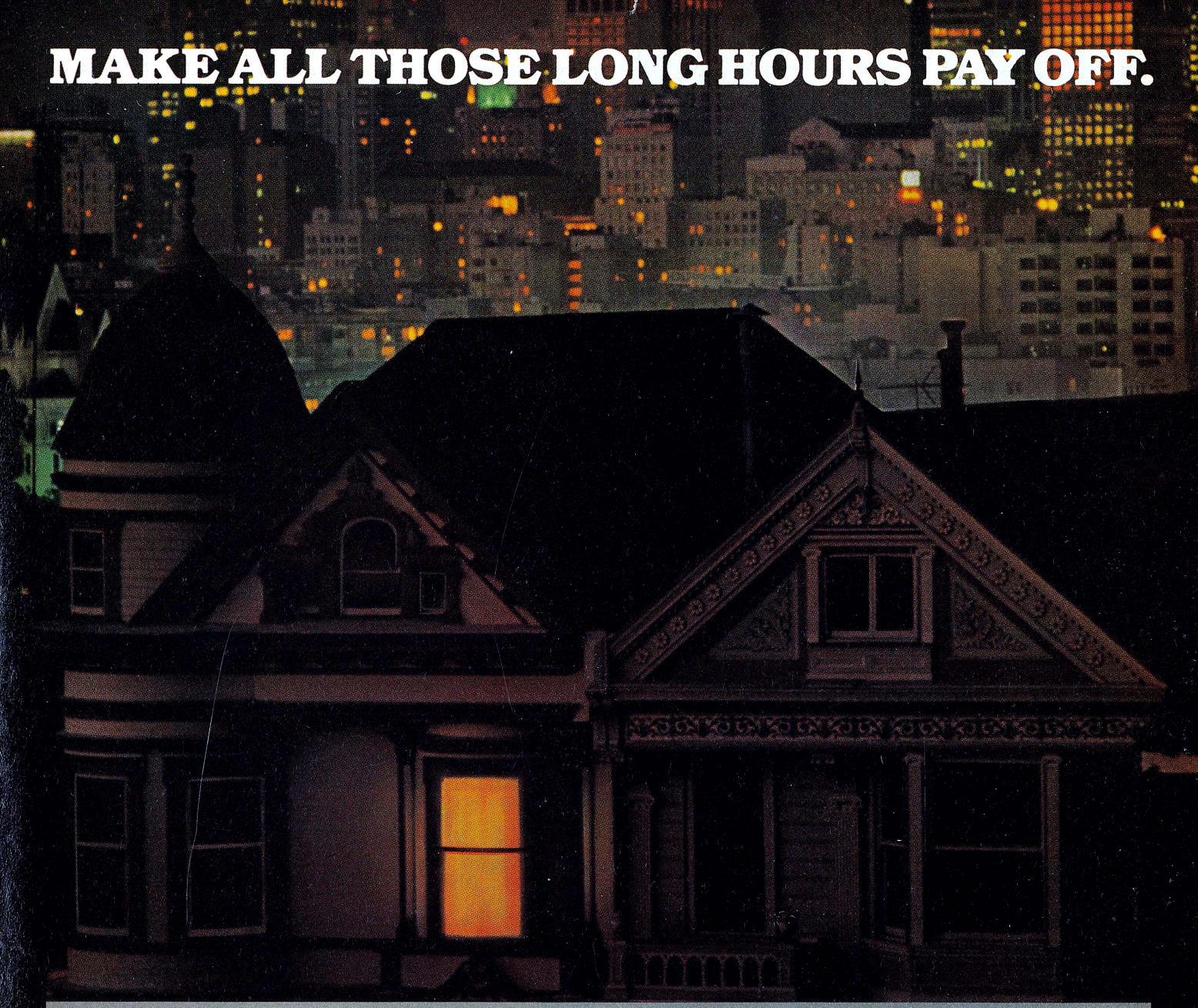
Budget/Tax Deduction Record--Keep track of tax deductions at the same time you are tracking your budget. **$39.95**

All programs include the Forth language.
* IBM PC only.

Available through your IBM PC Dealer or directly from: **I.C.**
13048 Olive Blvd.
St. Louis, MO 63141
(314)878-8838

‡WordStar & MailMerge are trademarks of Micropro Inc.

MAKE ALL THOSE LONG HOURS PAY OFF.



Now that you've written a better program, how do you get the world to beat a path to your door?

The demand for microcomputer software is one of the greatest business opportunities of the century.

That's why you spend those long, lonely hours at your computer.

But a successful software product needs more than inspiration. It needs documentation. Testing. Advertising. And, above all, distribution.

A great author needs a great publisher.

You can't do it all yourself. That's where Max™ Software Publishing comes in.

Max packages, documents, distributes, and supports software from independent authors just like you. We know what the IBM-PC market wants so we can help tailor your product, giving it the best chance to become a "best seller."

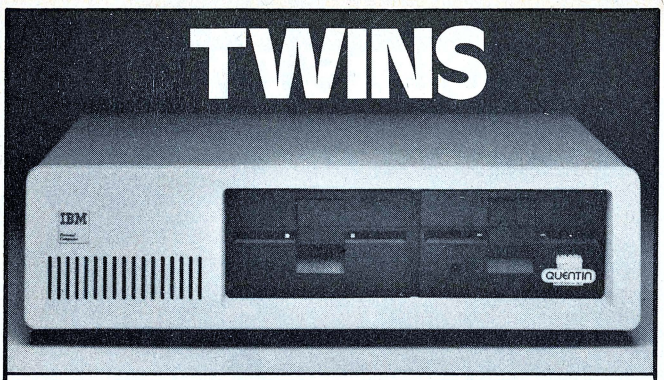
Max is publishing programs ranging from compilers to terminal emulators to utilities to business applications to games. All professionally packaged and documented. And all targeted for commercial success.

If there's a potential best-seller in *your* program, we want to hear from you. Send a brief description of your program or idea, and enclose a stamped, self-addressed envelope so we can get back to you.

Make all those hours pay off. Take your program to the **MAX.** (And if you're an IBM-PC user just looking for a great program, write for our free software catalog. Dealer enquiries invited.)

MAX™
SOFTWARE PUBLISHING

MAX Software Publishing, 2334 Walsh Road, Suite D, Santa Clara, CA 95050 (408) 727-2273 A division of Willoughby Corp.



Disk Drive Add-Ins From Quentin

Team up with Twin-Set™ from Quentin and go for the extra points in disk drive capabilities. Twice the storage, faster access, and Twin-Set's high quality Lead Screw Positioner will put your IBM PC ahead of the game.

Twin-Set is fully software transparent with IBM DOS and CP/M, and easily slides into the basic chassis. It is built around a reliable Siemens system with over 10,000 lifetime hours. MTBF 8500 power-on hours. On Track Head Seek virtually eliminates retries and copying errors, and diskette protection system guards disk hubs.

You waited for IBM...now add the components that will give you top performance. Special introductory price $249.00. Full one-year factory warranty. Count on Quentin for Quality... in disk drive systems.

To order: Check, Visa or Mastercard number. Calif. residents add sales tax. Call for specifications. Club discounts available.

QUENTIN RESEARCH, INC.
19355 Business Center Dr.
Northridge, CA 91324
(213) 701-1006
Telex: 910-493-2174

Winchester Disk Drive From Quentin

Faster than a speeding floppy. More powerful than a dozen diskettes. Able to store and retrieve 5, 10, 15, even 20 megabytes of formatted data on a single drive.

It's Big Brother, the IBM PC Winchester Disk Drive from Quentin. **Fast.** Accelerates from 0 to Read in 70 MS. DMA data transfer rate zips along at 5 MBits per second. **Powerful.** Compatible with IBM DOS, CP/M and protected software. 20 megabytes equivalent to 140 floppies. **Reliable.** Flick a switch and Big Brother's Self Test Diagnostic system checks all Winchester operating functions, verifying the integrity of the drive and the data. 32 bit Error Correction Code overrides all data field errors. MTBF 10,000 power-on hours, with no preventive maintenance required. Full one-year factory warranty. **Affordable.** 5MB $1995. 10 MB $2195. 15 MB $2395. 20 MB $2595. Count on Quentin for Quality... in mass storage subsystems.

To order: Check, Visa or Mastercard number. Calif. residents add sales tax. Call for specifications. Club discounts available.

QUENTIN RESEARCH, INC.
19355 Business Center Dr.
Northridge, CA 91324
(213) 701-1006
Telex: 910-493-2174

Cell	Format	Formula
C7:	Left	1+(C4/100)
B13:	Left	
B14:	Left	1+B13
B14:		/R:B15.B33R
C13:	Dollar	+C9
C14:	Dollar	+C7*C13
C14:		/R:C15.C33NR
D9:	Dollar	@CHOOSE(C5,C14.C33)

The formulas shown here are for use in *VisiCalc*. If you're using a different spreadsheet program, you'll need to translate the two /R formulas and the @CHOOSE. The first /R replicates the formula at B14 into the range B15 through B33; all formula references are relative. The second replication copies C14 into the range C15 through C33, holding the reference to C7 as a constant. The Choose function examines the contents of C5 and selects the appropriate value from the range C14 through C33. In our example in figure 1, cell C5 holds the value 5, so the Choose function returns the fifth member of the series C14 through C33—in this case $29,386.56.

Now that you have your spreadsheet, you can simply change one or more of the three variables—inflation rate, years to retirement, and today's need—and get an instant picture of your projected needs for retirement. Once you've determined your needs, you can begin to consider in detail how you might be able to attain that goal. For that task, a well qualified financial investment planner is a definite plus.

Next month we'll take a look at how you can graphically display some of this same sort of information using bar charts—even if you don't have a graphics printer or any tool other than a basic electronic spreadsheet. ▲

	A	B	C	D
4		INFLATION RATE:	8	
5		YEARS TO RETIRE:	5	
6				$ NEEDED
7		ANNUAL MULTIPLE:	1.08	TOMORROW
8				
9		TODAY'S NEED:	20000	29386.56
10				
11		FUTURE	$ NEEDED	
12		YEAR #	TOMORROW	
13		0	20000.00	
14		1	21600.00	
15		2	23328.00	
16		3	25194.24	
17		4	27209.78	
18		5	29386.56	
19		6	31737.49	
20		7	34276.49	
21		8	37018.60	
22		9	39980.09	
23		10	43178.50	
24		11	46632.78	
25		12	50363.40	
26		13	54392.47	
27		14	58743.87	
28		15	63443.38	
29		16	68518.85	
30		17	74000.36	
31		18	79920.39	
32		19	86314.02	
33		20	93219.14	

Figure 1.

Now even hunt-and-peck typists produce perfectly formatted letters...memos...presentations... scripts...*automatically.*

POWERTEXT® for your IBM PC: Announcing the next generation in micro-based word processing.

"A business professional wants to think about what he or she is writing. Not worry about getting the words in the right places on the printed page.

"That's why we developed a new kind of high-performance system that handles printed style automatically.

"PowerText knows exactly how you want your letters to look...your memos...reports...every kind of document you write.

"You define the formats. PowerText stores them in its style files.

"When you write, just bat out your copy on your IBM PC. Forget about printed style.

"Do all the editing you want.

"Then...just tell PowerText 'Make this a letter,' and you get a letter...'Make this a sales presentation,' and you get a sales presentation...

"It's almost like dictating to your computer. Even if you're an unskilled typist, you get a beautiful, perfectly formatted document...every time.

"And it's all very straightforward...very easy to use.

"We think PowerText is the way a word processor is supposed to work."

David P. Guest, President, Beaman Porter, Inc.

All this is automatic in every kind of document you write:

* Margins
* Indents
* Spacing
* Title page
* Envelope, label
* Intelligent page breaks
* Justification
* Centering
* Variable pitch
* Pagination
* Table of contents

Plus many other high-performance features:

* Complete editing capabilities
* Form letters
* Automatic headers and footers
* Boxed copy
* Print macros
* Up to 14 columns
* Automatic footnote numbers
* Boilerplate inclusion
* Vert. and horiz. border lines
* No limit to document length
* 132-character lines
* Superscripts and subscripts
* ...and much more

25% more data per disk

PowerText accommodates 200K bytes per disk, 25% more data than the 160K bytes accommodated by standard IBM software.

System requirements

PowerText is a run-time, bootable system. It requires at least 64K of memory, two disk drives, and a printer. It supports both serial and parallel printer interfaces, extended memory, and RAM disk.

Full 5-year warranty

The system is warranted for a full 5 years. Should it ever fail to perform as specified, Beaman Porter, Inc. will fix it at no charge.

Available from selected dealers
Or order directly from Beaman Porter.

VISA and MasterCard accepted. NY State residents add appropriate sales tax.

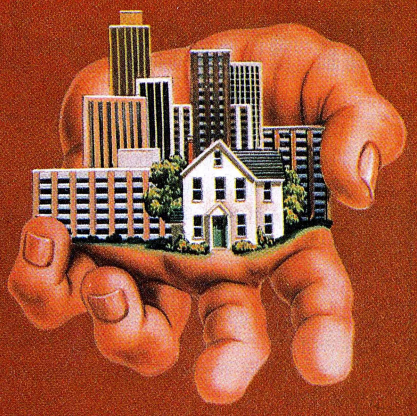
Complete software $399
Includes diskette plus excellent manual/tutorial
Manual/tutorial alone is $25

PowerText®

BEAMAN PORTER, INC.
High Performance Computer Products

Pleasant Ridge Road, Harrison, NY 10528 (914) 967-3504

"Real Estate deals may look the same. R.E.I.P. separates losers from winners."



Outside of money, our **R**eal **E**state **I**nvestment **P**rogram (R.E.I.P.) is the most valuable tool you can have at your command in real estate. And that's true whether you are a *beginning investor or a veteran.*

It provides you with the kind of information, guidance and analytical capabilities that help you decide which of two or more similar deals is the one to consider — what type of leverage may fit your situation best — whether a particular offering is a real opportunity or a disguised loser. In other words, R.E.I.P. is designed to give you the insight and expertise that was intuitive to the millionaires who founded huge real estate empires throughout history . . . before the time of the Apple II* or the IBM-PC.

With accurate analysis of a property's down payments, amortization, expenses, cash flow, tax consequences and other inter-related information, the knowing investor can make money in boom times or busts . . . in inflation, deflation or stagflation. And that's what R.E.I.P. is all about. It takes your input in a simple, logical and easy to enter way . . . probes for the necessary facts . . . scrutinizes the data and then automatically shows you the possibilities. Thus, you can examine the data, digest the variables, consider the potential payout against your personal goals and financial situation . . . and make a more informed decision, one not merely colored by emotion or someone's fast talk.

While there are other real estate software programs available, we believe you'll agree they fail to be competitive with R.E.I.P. Even though others cost more, R.E.I.P. contains many more features and provides the sophistication needed to operate successfully in today's challenging market.

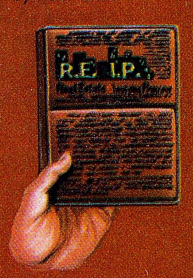
For example, R.E.I.P. is menu operated (so you save time and effort) — R.E.I.P. provides immediate interactive error checking — R.E.I.P. supports the newest real estate wrinkle, creative financing — and R.E.I.P. allows you to define negative amortizations.

Furthermore, R.E.I.P. is far ahead by having automatic report formatting . . . and by permitting multiple methods of income entry. Plus its reports are professional, as they are based on established Real Estate industry practices, with all information for an investment decision presented.

So, if you are engaged in real estate investing, or about to be, let R.E.I.P. work as your partner. It's **the** program for any investor . . . a **must** program for the sophisticated investor. Complete, on disk, with thorough, easy to follow documentation, it's priced at only **$129.95.** Specify the Apple II or IBM-PC version, at your computer store or from:

DATAMOST INC

9748 Cozycroft Ave.
Chatsworth, CA 91311 (213) 709-1202



*Apple II is a trademark of Apple Computer, Inc.

tradetalk

Δ Seventeen more retail outlets have become authorized IBM dealers, bringing the total to ninety independent stores in the United States. More are expected.

Δ **Epyx/Automated Simulations** (Sunnyvale, CA) has moved to a new, 14,000-square-foot building. The company's previous office was 5,000 square feet—and the original one, three years ago, was about 1,000 square feet. The firm's new address is 1043 Kiel Court, Sunnyvale, CA 94086.

Δ **Corona Data Systems** (Westlake Village, CA) has contracted with **Management/Personal Computers** (London, England) for the distribution of Corona's hard disk drives in England. Under the agreement, M/PC will also have the right to manufacture drives for the IBM in the future. Disk shipments in the first year will reportedly have a value of $2.8 million.

Δ **John Reutter III** has been named president of **Megasoft** (San Jose, CA). Reutter was formerly director of software development at Dynabyte. Meanwhile, Megasoft has opened a southern California office in Garden Grove, and **Thomas A. Chandler** has been named regional manager.

Δ The **American Management Association**'s self-study courses will be made available on database from **Management Contents** (Northbrook, IL). Also available is AMA's periodical, *Executive Skills*. The MC database currently indexes and abstracts articles from more than five hundred business publications, as well as books and proceedings of professional associations. It is available from Bibliographic Retrieval Service, Systems Development Dialog Information Services, and Datastar in Europe. A subset of twenty-seven business journals is available on the **Source.**

Δ **Spinnaker Software** (Boston, MA), a newly formed company, has signed with two design companies, **DesignWare** (San Francisco, CA) and **Computer Learning Connection** (Cambridge, MA), for a line of entertainment and educational programs for children. Initial release is scheduled for this fall.

Δ **Reader's Digest** has appointed **Roger A. Miller** as vice president of sales and marketing for the **Source.** Miller is a thirteen-year employee of Reader's Digest, serving previously as a circulation manager and later as business manager of the firm's books and recorded music division.

Δ The tax department of **Peat, Marwick, Mitchell & Co.** (New York, NY) has purchased a hundred Personal Computers for use in its local tax offices. The machines will be used for *VisiCalc* models, communications, tax return processing, word processing, and database management.

Δ **Barry Richman** has been appointed general manager of **Osborne/McGraw-Hill** (Berkeley, CA). He succeeds **Dr. Adam Osborne**, founder of Osborne/McGraw-Hill and inventor of the Osborne computer. Richman was previously publisher of Osborne/McGraw-Hill, and before that was manager of business and professional services in the company's professional and reference book division. ▲

Think About It

"The man who utilizes every minute of every hour becomes a bigger, better being every minute." Thus spake Thomas J. Watson, the man who made IBM what it is today.

Watson encouraged leadership on a one-to-one basis. A leader was expected to help people solve problems, not tell them what to do. We're trying to keep this philosophy alive at Softalk Publishing. We can't tell you how to solve all your personal computing problems, but we can steer you in the right direction.

Although Watson might wince at the notion, we also believe in giving our magazines away free for the first year. If you own a pc and have never received *Softalk for the IBM Personal Computer*, send in your name, address, and pc serial number and we'll start sending you the magazine free of charge.

Send us the name of a friend who is not already receiving *Softalk for the IBM Personal Computer* and you'll get a special bonus. Mail in your friend's name, address, and pc serial number and we'll send you the back issue of your choice free of charge. Such a deal! The first three issues (June through August 1982) are still available, but they're going fast.

Remember, every minute of every hour counts, so sign up today. We believe you won't waste any minutes or hours reading *Softalk for the IBM Personal Computer*.

softalk
for the IBM Personal Computer

11021 Magnolia Boulevard
Box 60
North Hollywood, CA 91603

exec
SOFTECH
MICROSYSTEMS

The p-System Purveyors

by Allan Tommervik

One of the more interesting and useful byproducts of the introduction of the IBM Personal Computer has been renewed emphasis on operating systems for desktop computers.

Most end users have a tendency to take the operating system as a given: it's there, it does what it's supposed to, and they can't do anything about it anyway.

With the advent of the pc, that situation changed. Users were offered three operating systems, and it behooved them to bone up on the merits of each. So far, most users are opting for IBM DOS by Microsoft; but both Digital Research's CP/M-86 and SofTech's p-System are finding their adherents.

Three Coins in the Fountain. There's been a subtle difference of approach in IBM's marketing that's been healthy for all three companies. Most manufacturers have presented the end user with an operating system as fait accompli, take it or leave it; though, for those who chose to leave it, a second system was sometimes available at additional cost. IBM has sanctioned all three systems by emblazoning their initials on each package. This gives the contenders equal standing and lets the marketplace be the ultimate judge. For novices, this may be an unwelcome burden; for more experienced users, it's a welcome change.

For the three publishing companies involved, it's been a change also. CP/M has been first choice for many machines and second for most of the rest. The pc marketplace is one of the few in which Digital Research has the opportunity to test its product in a truly competitive environment.

Microsoft hadn't marketed an operating system prior to IBM DOS, so being given equal standing with the established products has helped DOS find its niche. Its actual standing relative to the others won't be measurable for at least a couple of years.

But, of the three competitors, the p-System should get the biggest boost from the raised consciousness of the end user. It's been around for a while without making too many waves. Instead, SofTech has approached the market with a deliberate, planned attack on the manufacturers, rather than the end users. The approach to manufacturers has been on target; now it's time to increase the visibility of the product to the end user.

Increasing visibility may be overstating the case. To many end users, the p-System is only its language—Pascal; many others have roughly the same understanding of the attendant operating system as they have of quantum mechanics.

SofTech's marketing plan deliberately placed the emphasis elsewhere for the first two years, so the company's anonymity among end users is understandable. Now the p-System is coming out of the closet, so to speak; and it's now that the results of years of dedication and sacrifice on the part of many contributors will get the ultimate test.

Origin of the Species. This month's cover story on Kenneth Bowles describes the evolution of Pascal and the p-System at the University of California at San Diego. While the effects of his efforts continue to be felt at all levels of computing, his work at UCSD was only the first chapter in the story of the p-System. When Bowles was asked to select a company to whom the university could entrust the product, he chose wisely and well.

SofTech took the cream of the university crop, leavened it with seasoned management from its own ranks, and continued the effort begun at the university. With one exception, that initial SofTech management team remains in place as they vie with Digital Research and Microsoft for market share.

The actual company that acts as the steward of UCSD Pascal and the p-System for the university is SofTech Microsystems, a wholly owned subsidiary of SofTech. The parent is a large Massachusetts-based software concern with major government projects. The subsidiary's presence in southern California at the very time the university was looking for a trustee was a fluke.

SofTech had on its staff in the late seventies an iconoclastic Californian named Al Irvine. Irvine had a background in big systems, including a stint directing software engineering activities for a division of National Cash Register; but, with his quick wit and wide-ranging interests, he belied the stereotype of the mainframe programmer. In prior days, when programs were represented by as many as ten thousand punch cards, meticulousness, rather than imagination, was the coin of the realm; and most programmers had a personality to match.

Irvine, on the other hand, had broader interests, one of which was increasing the productivity of the programming fraternity. SofTech had the same interest, which is what moved Irvine to uproot himself from California and head for Boston. Eventually, common interest or not, Irvine felt himself drawn back to southern California and tendered his resignation.

A Place in the Sun. SofTech was familiar with the refrain that a good man is hard to find. So instead of accepting Irvine's resignation, the company created a branch office in the San Diego suburb of Escondido and offered him the post. It should be understood that at that moment SofTech needed a southern California branch with approximately the same urgency as *Homo sapiens* needs a second appendix. That situation was soon remedied.

Even as he was setting up the branch office, Irvine continued to pursue promising leads that might result in tools to increase programmer productivity. Eventually, the UCSD project on Pascal came to his attention and he called Bowles. Irvine and Bowles had been acquaintances for some years, and Bowles had already arrived independently at the thought that SofTech might make a good home for his project.

Bowles outlined Pascal and the p-System and the problems the university was having with the IRS. Irvine relayed the information to SofTech president John Brackett, who encouraged Irvine to enter into negotiations with the university. From those talks came the agreement to entrust the p-System to SofTech.

It was during this time that Brackett's influence began to be felt. Brackett was one of the original founders of SofTech and to this day remains on the board of directors. He saw the promise of portability as a means of increasing programmer productivity, and he recognized the potential of the microcomputer. He pushed for the creation of SofTech Microsystems as an independent subsidiary and then went the extra mile.

Royal Flash. Dr. John Brackett resigned his post as president of SofTech to become the president of SofTech Microsystems. SofTech is a large, publicly owned company, so his action seemed on a par with Queen Elizabeth abdicating the British throne to be-

Opposite page: the p-team. John Splavec, president (top left), has brought more aggressive marketing approaches and a wider market perspective to his new company; Al Irvine, vice president of engineering (top right), made the contact that brought the p-System to SofTech; Julie Erwin, vice president of marketing (bottom left), started with the system at UCSD and is still its foremost spokesman; and Mark Overgaard, manager of advanced development (bottom right), was the grad student most responsible for implementing the p-System design.

come governor of the Falkland Islands. And for a while the change must have seemed that drastic.

Brackett went from a company with plenty of resources to the equivalent of a start-up company where the term *cash flow* was a euphemism for expenses. Early on, SofTech Microsystems demonstrated a real capacity for spending money without a commensurate talent for earning it. When that caused the customary concern in high places, Brackett became the most ardent advocate of the subsidiary and its economic potential.

He was architect of the early marketing program that called for the emphasis on proselytizing the hardware manufacturers. He was a tireless laborer in that cause, keeping a schedule that sometimes resembled that of a door-to-door drummer as he went from major manufacturer to major manufacturer extolling the virtues of Pascal, the p-System, and portability. The success of his missionary work is reflected today by the fact that the p-System is available on every major microprocessor and on every widely used microcomputer.

How Scarce Are They? Executives with the ability to function in resource-poor environments are nearly as scarce as hen's teeth. Equally rare is the executive who recognizes when his company has reached a point where it will benefit from leadership of a kind different from his. Brackett is that kind of exec, and earlier this year he stepped down as president of SofTech Microsystems to make way for John Splavec. Splavec's different vision of the marketplace has already had a galvanizing effect on the company.

SofTech has been able to respond rapidly to Splavec's new approach because of the care with which Brackett crafted his management team. From SofTech *pere*, he collared Irvine and made him vice president of engineering. From Ken Bowles's group at UCSD, he selected Julie Erwin and Mark Overgaard.

Erwin is now vice president of marketing and Overgaard is manager of advanced development. Both held analogous positions in the UCSD group that started this whole thing. Neither had exactly this kind of project in mind when they started their college educations.

Overgaard majored in physics and caught the computer bug while at Geneva College in Beaver Falls, Pennsylvania. While there he began helping one of the faculty members, an old family friend, feed punch cards into an 8K IBM 1130. He got hooked on computers and spent as much time in the college data processing department as he could.

But he did not enroll in UCSD's computer science department when he entered grad school, because he didn't feel that computer science was an end in itself. Needless to say, Overgaard soon shed that shibboleth, as he'll wryly admit.

There was a tremendous enthusiasm and excitement in the computer science department at UCSD, and Overgaard soon focused his efforts in that direction. He wrote a paper on memory management that his professor suggested he show to Ken Bowles. Bowles invited Overgaard to take a look at a little project he had going. Of course, the little project was Pascal development, and Overgaard immediately signed on.

Pascal Is a Many Splendored Thing. This fellow with the low opinion of computer science soon was the key student working on the design and implementation of Bowles's concept. Overgaard concedes that, during that period, Pascal was his only interest.

After an article by Bowles appeared in *Byte*, UCSD was flooded with requests for further information, and a team of student volunteers was mobilized to respond. Julie Erwin was among them.

A math major at the University of Arizona who was finishing work on her degree at UCSD, Erwin was one of those students who throw themselves into extracurricular activities. At UCSD, she'd gotten involved in a student co-op venture and in the course of that activity had met Ken Bowles. She served as proctor for one of his classes and later picked up a job as staff assistant in the computer science department.

The same sense that recognized that a student co-op would fill a void led Erwin to recognize gaps in the effort to manage the Pascal project. She began picking up the various pieces and became an integral part of the effort.

The original SofTech Microsystems organization was composed of only six persons. By terms of the original contract between the company and the university, Overgaard spent some time in each place, smoothing out the transition, before joining SofTech full time. When the university turned the project over to SofTech in 1979, version II.0 was the product. Now SofTech is selling version IV.0, so the last three years have been active ones.

From that original staff of six in Escondido, the company has grown to more than ninety employees and this month will leave its present quarters for a 33,000-square-foot facility in Rancho Bernardo, California.

The product has seen similar expansion. The original impetus of the project was to implement an extended version of Pascal. But the pseudomachine developed to interpret the compiled Pascal code has proven sufficiently versatile that Fortran and Basic have been added to the p-System.

Young Clucks. Under Irvine's guidance, a print spooler was added. He chuckles when he recalls the questioning looks he got when that task was defined. Irvine's staff is mostly composed of younger engineers lacking his mainframe experience, and they didn't immediately recognize the value of the application. Now, as they use it, they appreciate the value of spooling.

From the beginning, the development of the p-System has been a recursive exercise, with Pascal being used as the development language at the same time it was being defined and extended. Irvine points to this recursion as one of the strengths of the system. SofTech uses the p-System for all its development activities, which means the system is tested daily. Whenever one of Irvine's engineers comes up with a better method of handling some task, it's implemented into the commercial system as well as the in-house one.

What Ken Bowles sought when he started his project was port-

PEP

BUSINESS SYSTEM

Compare our software to the competition

1) Inventory Control
BILLS-OF-MATERIAL — PURCHASE ORDERS; SALES ANALYSIS — STOCK LABELS + MORE

2) Accounts Receivable
ORDER ENTRY & PACKING SLIPS; INVOICES FROM O/E OR POINT OF SALE; CUSTOMER PAYMENTS W/DISCOUNTS; STATEMENTS; AGED RECEIVABLE REPORTS + MORE

3) Accounts Payable
VENDOR INVOICE ENTRY AND CHECK WRITING; BANK DEPOSIT, AGED PAYABLES REPORTS + MORE

4) Payroll
DATA ENTRY, AUTOMATIC CALCULATION & DEDUCTIONS, PAYCHECK WRITING, MO-QTR-YTD REPORTS — 941-A and W-2

5) General Ledger
FULLY INTERACTIVE WITH ALL OTHER PROGRAMS! TRIAL BALANCE, REVENUE, EXPENSE, OPERATING & BALANCE SHEET.

Regular $795 —Special— $495.00

For Free Brochure, Call or Write:

PERFORMANCE ENGINEERED PROGRAMMING

3970 Syme Drive
Carlsbad, CA 92008 (714) 434-6023

ability—the ability to move software easily from one system to another. What SofTech believes they have, and what they're now trumpeting, is universality.

The battle cry of universality is not directly aimed at the end user. At the moment a pc owner inserts a disk in the disk drive and boots it, the only question is whether it will run or not. Whether umpteen thousand other microcomputer owners can do the same thing or not is of little import at that instant.

It's here that the question of programmer productivity becomes central. If a programmer must choose to write for only one computer, or one common subset of computers, there are two direct ramifications. One is the limited return that programmer will get for his effort. The other is that owners of those computers outside the pale of his effort are deprived of his contribution.

Plastic Man. If the object code is truly portable, the programmer will be maximizing his reach in the marketplace and therefore maximizing his possible return. In theory, when the programmer can maximize his reach, he can afford more research and development and turn out a better and more responsive product.

So the question of whether a given disk will boot is the microview. The question of whether there's even a disk there to insert in the drive is the macroview. Universality addresses that question by attempting to persuade software authors and publishers that the p-System provides that kind of potential.

Listening to Erwin, it's hard to doubt. The litany of microprocessors on which the p-System runs is comprehensive. The list of computer brands using these processors is voluminous.

SofTech has addressed the universality question at the end-user level as well, with a piece of software they call the *Universal Medium Adapter*. The concept of the universal medium addresses the various formats used to write data on disk.

It's all well and good to talk about universality, but if the end result is still thirty-five separate packages to address thirty-five discrete markets with thirty-five different means of writing to disk, SofTech's message is watered down.

With the universal medium, end users on many kinds of computers can acquire applications software in the single universal medium format. The user will also need a *Universal Medium Adapter* program tailored to their computer. When that software is executed, it seizes control of the disk controller long enough to copy the universal medium disk to a disk in their machine's native format.

It's typical of SofTech's role as a trustee for the university that such a powerful concept isn't even looked at as a revenue source. While other publishers might have dollar signs in their eyes over such a versatile program, SofTech is pushing it at publishers as just another minimum cost service.

If the concept gets widespread acceptance, publishers will need to prepare only one package and computer stores will only have to carry that one package to serve many machines.

Meanwhile, Overgaard is addressing the problem of supporting larger memories. This is no trivial consideration for something like the p-System that stresses portability. As Overgaard points out, the question is how to maintain portability while providing the capability to use more than 128K for code and data.

A program written for a 48K Atari will take advantage of 128K on the pc without change. How to extend that range to 256K or more is a subject of priority at SMS.

While Irvine and Overgaard wrestle with the capabilities of the p-System, Erwin and new president Splavec are taking aim at new segments of the marketplace. Where the emphasis has been on manufacturers, now it's on publishers, retail outlets, and end users.

A Thousand and One Customers. Splavec came to SofTech from Informatics, a mainframe software and services supplier, and he's brought to the company an awareness of the Fortune 1000 market as well as the knowledge of how to penetrate that market. Splavec believes the p-System is the ideal product for large companies in that it frees managers to make hardware decisions with the sure knowledge that a common base of software will bridge the hardware differences.

Under his administration, SofTech has gotten more aggressive in their marketing to end users as well—taking on Apple Computer head to head in a campaign to convert thousands of Apple Pascal users to the p-System.

Erwin's taken up the challenge of wooing the software publishers to the p-System. Among her early successes has been the licensing of the system to VisiCorp for use on *VisiSchedule*. She believes the determination of which operating system will become the de facto standard for the pc will depend on which system is most successful in attracting publishers.

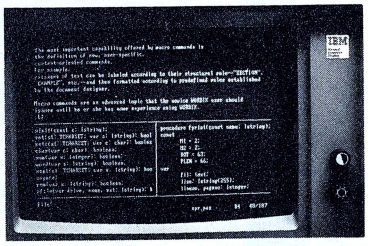
Notwithstanding SofTech's preoccupation with universal operating systems and attendant support, the company has developed a program especially for the pc. Tentatively called the *BIOS Splicer*, the software addresses a weakness Overgaard observed in all three pc operating systems—they ignore foreign peripherals and the needs of the manufacturers of foreign peripherals to develop drivers.

The Cutting Edge. The way all three operating systems are designed, it is a nontrivial effort to adapt the I/O codes to different devices. The *BIOS Splicer* will stitch the driver for a foreign peripheral to the p-System BIOS. Then the p-System Linker will stitch the mended BIOS to low-level programs. The splicer was designed for use by software publishers and peripheral manufacturers but will be available for end users as well.

As with the *Universal Medium*, SofTech is viewing the product not as a main source of revenue, but as a service to make microcomputing more accessible. And it's the pc owner who'll benefit.

The SofTech folks are a dynamic and effective mixture of youth and experience. And while they remain dedicated to proving the p-System to be the universal answer to operating systems, they've clearly got an eye out for specifics, as the *BIOS Splicer* shows. That's a fact that augurs well for pc owners. ▲

Edix™ + Wordix™ has word processing features that our competitors are still dreaming about . . .



$390 buys the whole dream.

Edix Editing Features
Up to 4 screen windows
Up to 12 file buffers
Top-to-bottom and side-to-side scrolling.
Regular expression searching and translating
Intra- and inter-buffer moving and copying
Online help
Online tutorial

Wordix Formatting Features
Auto footnotes
Auto table of contents
Auto hyphenation
Multi-line headers and footers
Multi-column layout
Powerful macros
Data file access

Call or write for our brochure.

Requires 128K IBM PC
Edix alone: $195

EMERGING TECHNOLOGY

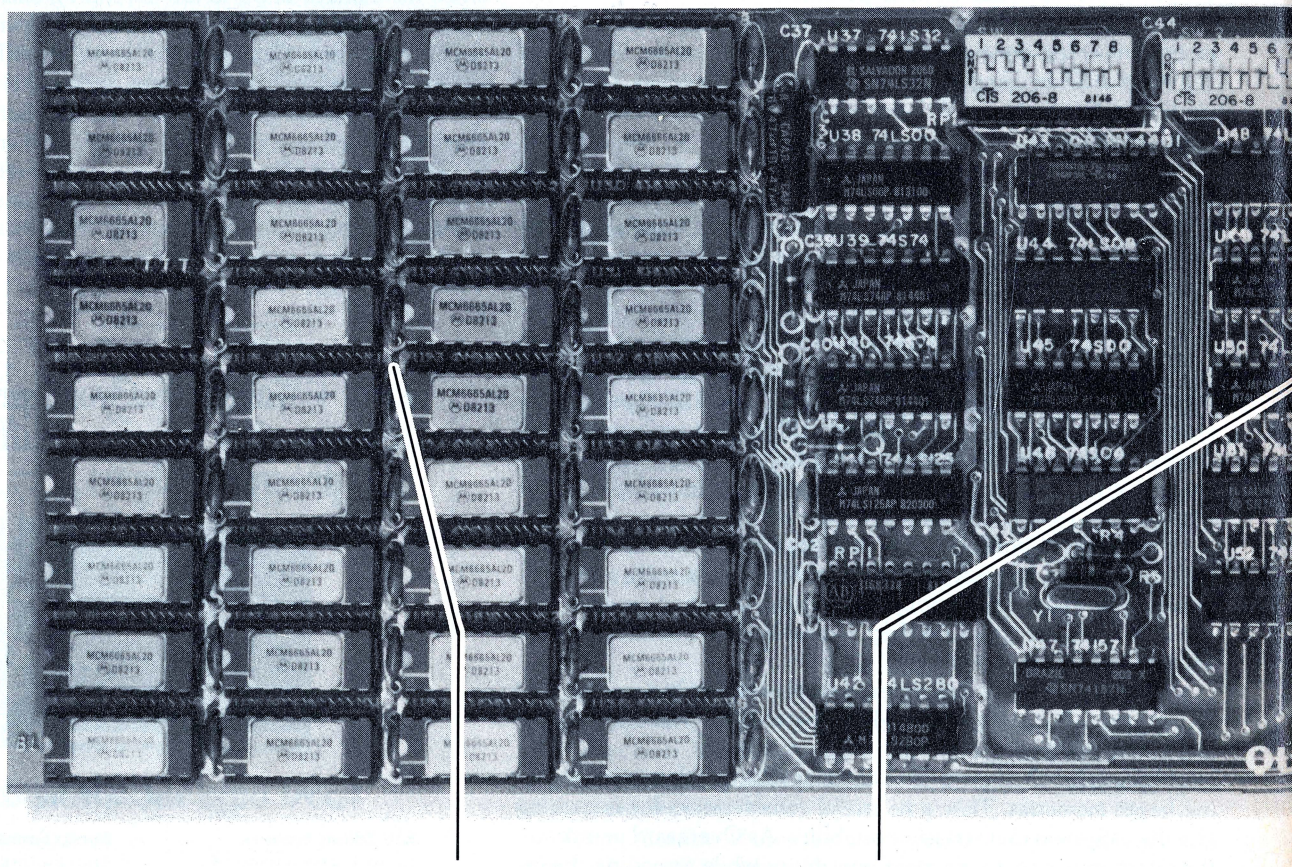
Box 154 Louisville, Colorado 80027 303 447-9495

TM Trademark of Emerging Technology Consultants, Inc.

QUADBOARD™

THE FIRST AND ONLY BOARD YOUR IBM PC MAY EVER NEED.

Your IBM personal computer is a very versatile piece of equipment. Perhaps more versatile than you realize. New applications and functions are being developed every day. Now with Quadboard by Quadram you can keep your options open for tomorrow's technology. Following in the tradition of Quadram Quality, four of Quadram's best selling IBM boards have been combined into one board. Your remaining slots will be left free and available to accommodate future expansion needs and uses which you may not even be able to contemplate today.



PROVEN DESIGN.
Quadram has been shipping IBM boards with each of the Quadboard functions on separate boards since December, 1981. They are still available as separates (including a Dual Port Async Board) for those who desire a quality board but do not need to keep slots open for future expansion. And they all come with a one year warranty from the leader in technology applications.

256K MEMORY EXPANSION.
Socketed and expandable in 64K increments to 256K, full parity generation and checking are standard. A Quadboard exclusive feature allows parity to be switch disabled to avoid lock-up upon error detection. The dip switches also allow it to be addressed starting on any 64K block so that it takes up only as much as it has memory installed. Memory access and cycle time naturally meet all IBM specifications.

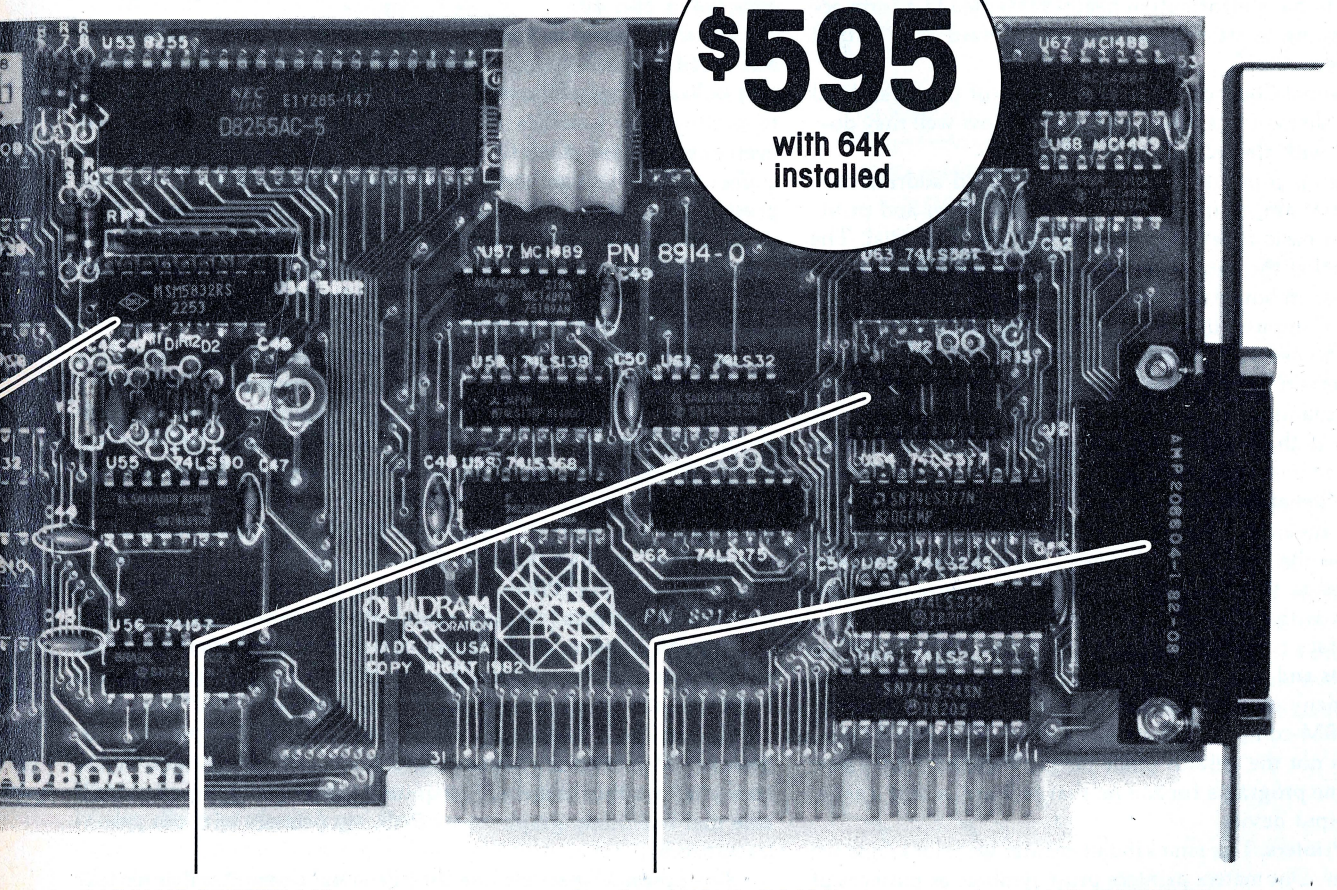
CLOCK/CALENDAR.
Quadboard eliminates the hassle of manually inputting the date on system boot-up by providing for the clock and all software routines necessary for inserting the appropriate programs on your diskettes. The internal computer clock is automatically set for compatibility with most software routines which utilize clock functions. On-board battery keeps the clock running when the computer is off.

BY QUADRAM

ALL ON ONE BOARD
Now you can utilize all the PC's capacity with Quadram's extremely flexible configurations. And it's totally compatible with IBM hardware, operating systems, and high level languages. It's a full-size board that can be inserted into any free system slot and it even includes a card edge guide for securely mounting the card in place.

INCREDIBLE PRICE!
Priced at $595 with 64K installed, $795 with 128K, $895 with 192K and an unbelievable $995 fully populated with 256K. Quadboard! You pay less for more and still protect the few limited empty slots in your computer.

$595
with 64K installed



PARALLEL PRINTER I/O.
A 16 pin header on Quadboard is used for inserting a short cable containing a standard DB25 connector. The connector is then mounted in the knock-out hole located in the center of the PC backplane. The parallel port can be switch disabled or addressed as Printer 1 or 2. No conflict exists with the standard parallel port on the Monochrome board. The internal cable, connector and hardware are all included.

ASYNCHRONOUS (RS232) COMMUNICATION ADAPTER.
Using the same chip as that on the IBM ASYNC board, the device is software programmable for baud rate, character, stop, and parity bits. A male DB25 connector located on the back connector is identical to that on the IBM Async Adapter. The adapter is used for connecting modems, printers (many letter quality printers require RS232), and other serial devices. Switches allow the port to be configured as COM1 or COM2 and the board fully supports IBM Communications Software.

AVAILABLE NOW!
Quadram does not advertise products until shipments have begun. Count on it. But order today for immediate delivery.

ASK YOUR DEALER.
All products are sold through local personal computer dealers. If yours does not stock Quadram, please ask him to call us at (404) 923-6666.

QUADRAM CORPORATION

4357 Park Drive/Norcross, Ga. 30093

BOARDS & BUSES

by Richard Kaapke

Mankind's finest invention in the service of information management is the electronic computer. The most prevalent means of information exchange, however, remains the printed page—as your reading of this article attests. Once equipped with a computer, anybody who wants to have information neatly presented in a form accessible to all is going to start thinking about a "hard copy output device," or printer.

The IBM Personal Computer can use a variety of printers. We'll look at several of them in this article and discuss how well they do—and don't—work with the pc.

Before taking up printers themselves, we should address ourselves to the connection, or interface, between computer and printer. There are two basic types of interface: parallel and serial. The distinction is whether the data goes out to the printer several bits at a time (parallel) or in single file, one bit at a time (serial).

The Personal Computer uses a parallel printer interface but may also be equipped to use a serial interface. Both parallel and serial interfaces follow certain standards that define the number of electrical wires each contains and the functions assigned to each wire. A parallel interface is the more complex of the two, because it uses more wires and imposes narrower tolerances on the signals each wire carries. The benefit of complexity is speed; parallel interfaces drive printers at their top speeds.

Parallel is also the default, or assumed, mode of operation for printers connected to the Personal Computer. All programming on the pc will work with compatible parallel interfaced printers.

A serial interface is much simpler than a parallel one, requiring fewer signal wires and having more tolerance for simple discrepancies. There are many more printers with serial interface capability than there are IBM-compatible parallel printers. The serial interface, however, is not the default mode on the Personal Computer, and there are some programs for the pc that will not accept a serial printer as an output device.

Dot Matrix Printers. The first kind of printer we'll look at is the dot matrix printer. Dot matrix printers print symbols as patterns of dots within a predefined box or matrix. The more dots available in the matrix, the better the image quality. A typical matrix size is nine rows of seven dots each; this is the matrix used by the IBM Personal Computer Printer and the Epson family.

Dot matrix printers may or may not have true descenders. This means the bottoms of the lower-case letters g, j, p, q, and y may or may not fall below the line the capitals stand on. If a printer doesn't have true descenders, you may find its lower-case alphabet and some of the other symbols hard to read. Be sure you see a copy of the complete character set produced by the printer in question.

Parallel printers guaranteed by IBM to work with all IBM-supplied software are a minority of one: the IBM Personal Computer Printer. This machine is actually an Epson MX-80 lookalike; the printer mechanism is made by Epson.

Epson makes a varied line of printers, and all of them function with the Personal Computer comparably to the one that bears IBM's name; in fact, there's nothing the IBM printer will do that the Epsons won't also do.

The most recent and popular members of the Epson line are the MX-80 III, the MX-80 F/T III, and the MX-100 III. The III designation indicates they all have the new Graftrax capability—the ability to produce dot graphics and print in any of twelve styles. The twelve styles are normal and slanted printing in each of six strike modes: single-strike, emphasized, double-strike, compressed, expanded, and combined compressed and emphasized.

The IBM Personal Computer Printer and the earlier MX-80s without Graftrax lack the ability to print dot graphics and slanted type. They offer the six strike modes, however, and the Graftrax feature is available as an option.

What's dot graphics? Remember that all symbols produced by dot matrix printers are made up of little dots. Graphics produced on screen by the Personal Computer are formed in the same way, although the size of the matrix may not be the same and in some modes the dots are colored. A printer with dot graphics capability can be controlled directly by the computer in such a way that a program can make a printed copy of a graphics screen or picture.

Because the IBM Personal Computer Printer doesn't have dot graphics capability, IBM designed the PrtSc function without giving it the capacity to send graphics to the printer. You can't ordinarily print graphics or, for that matter, special text characters (accented letters, Greek letters, border characters, and so on) using PrtSc, even if you have a printer with Graftrax.

IBM has no announced plans to enhance the PrtSc function, but there are a lot of enterprising programmers who do. In fact, by the time you read this, there will be quite a selection of programs available that will dump graphics, via PrtSc, to printers with dot graphics capability.

The Epson MX-80 and the IBM Personal Computer Printer take tractor-feed forms only (paper with punched holes on both left and right sides) and print with true descenders at a speed of eighty characters per second. All Epsons print bidirectionally, which means they don't occasion any delays while the print head returns from right to left. This is a big improvement over unidirectional printers.

The Epson MX-80 III and MX-80 F/T III are both portable and compact. Both have well-written user's manuals to help you understand how to operate the printer, add supplies, and use all the special features.

The MX-80 F/T has all the features of the MX-80 III plus the ability to handle single-sheet forms (letterhead, for example). The F/T has a tractor-feed mechanism that's easy to remove and reinstall. The F/T also has a rubber platen rather than a metal one, al-

lowing the printer to run more quietly. If you have a need to print on bonded paper, an MX-80 F/T III is an excellent choice.

The MX-100 III is the wide-carriage version of the MX-80 F/T III. Its size makes it less portable than the MX-80, although it's still fairly easy to move. The 100 can take a form 15 inches wide (compared to the 9½-inch limit of the MX-80) and has all of the MX-80 F/T III's capabilities. If you need a wide spreadsheet printout or simply have program or data listings that require those formidable 15-inch forms, the MX-100 III is a good choice.

Another dot matrix printer known to work with the IBM Personal Computer is the Anadex DP-9501, which has most of the Epson MX-100 III's capabilities—except for the single-sheet feed—and many of the 100's print styles. The DP-9501 prints a lot faster (faster than 200 characters per second) and seems to be quieter than the Epson. It will print graphics, but it's uncertain whether a graphics printout program will be made for it.

You may run paper into the Anadex either straight up from the bottom (if you use a special printer stand) or from the back (if you don't). The Anadex requires that the wire on the IBM parallel interface cable leading to its pin number 36 be cut before you use the printer—but this doesn't impair the operation of the printer in any way.

Most of the printers known to work in parallel mode with the Personal Computer have what's called a Centronics-style parallel interface. However, not all printers manufactured with Centronics interfaces work with the Personal Computer. It's best to ask before you buy.

Letter-Quality Printers. If it bothers you to see printouts that look as though they were made from a computer rather than from a typewriter, you might consider a letter-quality printer. Typically, a letter-quality printer uses a print element (of which you often have a selection) and prints much more slowly than a dot matrix printer; a typical top speed for letter-quality printers is about fifty-five characters per second. As compensation for slowness, however, you get a printout that looks like it was typed—or even typeset.

Letter-quality printers all have true descenders. They also offer the option of printing in pica, elite, or fifteen pitch; some even do proportional spacing. Proportional spacing is what you see in a typeset document, where characters vary considerably in the width allotted to them.

Some letter-quality printers are much more intelligent than their dot-matrix cousins. Some offer automatic justification (the even left and right margins you see here) as well as boldface and underscoring. This is to be distinguished from the kind of underscoring and emphasis that, for example, the Epsons are capable of; the Epsons and other printers of their class require a second pass of the print head to achieve underscoring or boldness, whereas the intelligent letter-quality printers can backspace within a line.

Letter-quality printers known to work with the IBM Personal Computer include the Brother HR-1 and the C. Itoh F10 StarWriter. Both of these have Centronics interfaces, and both produce a fine, typewritten copy. The C. Itoh retails for about $700 more than the Brother, but it's faster (forty-five versus sixteen characters per second) and has the following features missing in the HR-1: half linefeeds, proportional spacing, automatic justification, and a graphics mode.

The F10 also uses Diablo printer supplies (for the Diablo 630 or equivalent), which are typically less expensive than supplies for the HR-1. Both the HR-1 and the F10 have an optional add-on tractor-feed mechanism, which is helpful for the printing of form letters and long documents. C. Itoh has a faster model called the PrintMaster, and the PrintMaster, although more costly, may be ideal for an application where speed is essential.

Serial Printers. Serial printers include all the aforementioned machines (all either have serial interfacing as an option or have built-in dual interfaces—serial and parallel), plus many others. The Diablo 630 is a serial-interfaced, letter-quality printer without a parallel option (Renaissance Technology makes a converter cable that connects a Diablo to a parallel port). Another serial-interfaced printer worth mentioning is the NEC SpinWriter.

The Diablo 630, by Xerox, is a large (and heavy) letter-quality printer. The machine is made domestically, and there are many Xerox repair facilities available. Diablo has been a standard setter in letter-quality printers for years, and there are many computer programs that take advantage of the Diablo's special printing capabilities. In fact, many of the newer letter-quality printers offer a "Diablo compatibility" mode.

The Diablo can do most of the things the F10 can do—although you have to pay extra for automatic underlining, boldface, justification, and graphics. The 630 prints at forty-five characters per second and is the only printer that uses both metal and plastic print elements; all the others use only plastic. Metal makes a harder and better impression, and the print element lasts much longer.

What Kind Do You Need? The list that follows may help you decide what features you need. If you want the capability listed on the left, you need the feature on the right:

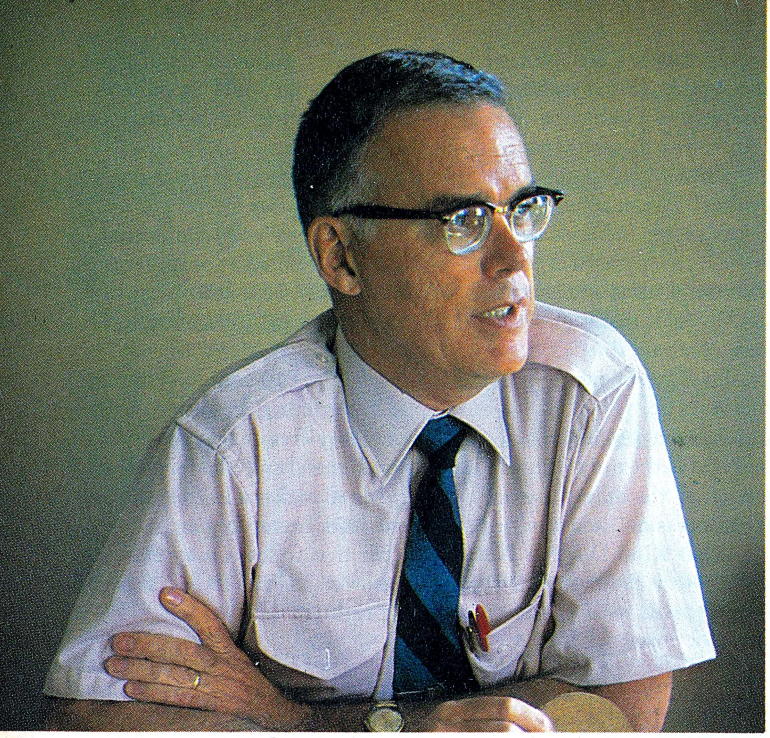
- Use of single sheets of paper (letterhead, for example) — Friction feed platen
- Printout of pictures drawn on screen (charts and other graphics) — Graphics
- Superscripts (trademark signs, exponents) — Negative half linefeed
- Subscripts (chemical formulas, for example) — Half linefeed
- Printing in either direction — Bidirectional printing
- Use of continuous-feed perforated paper — Tractor feed
- Typewriter-like printout — Letter quality
- Quality comparable to typesetting — Proportional spacing

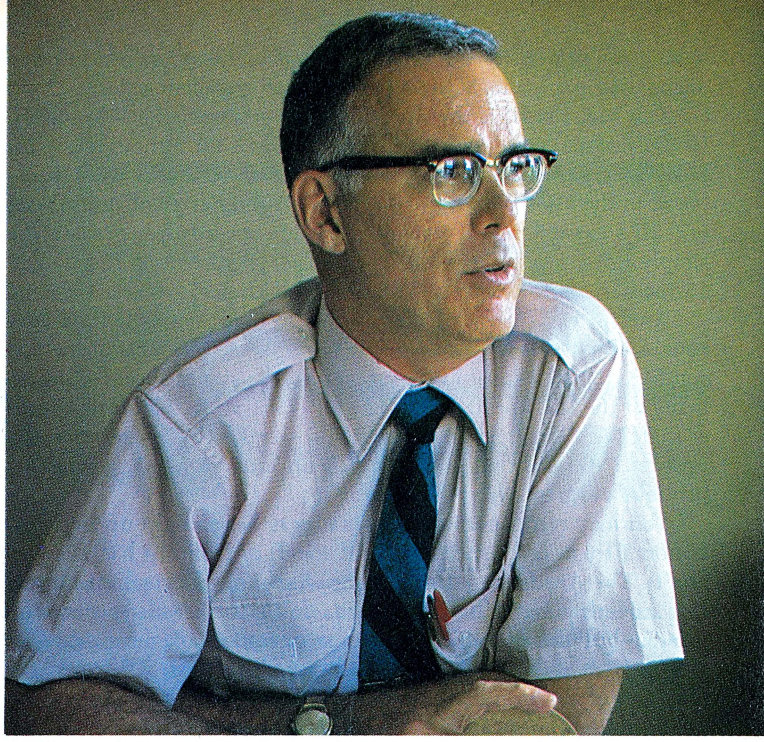
Table 1 summarizes the features of the printers discussed in this article.

A footnote about ribbons: all the letter-quality printers can use carbon film ribbons, which last only for a single use but produce the cleanest and sharpest possible printout. The HR-1 can only use carbon ribbons; all the others can also use cloth ribbons made into a continuous, reusable loop. Cloth ribbons give you more print for your dollar. The best buy in a carbon ribbon is a multistrike ribbon that can be struck many times and will advance slowly after each strike. Typically these cost more than standard carbon ribbons, but they last up to ten times longer. ▲

Name	Price $	Carriage Width	CPS Speed	Letter Quality	Graphics	Friction Feed	Half Linefeeds	Tractor Feed	Proportional Spacing
MX-80 III	645	9"	80		X			X	
IBM pc Ptr.	595	9"	80					X	
MX-80 F/T III	755	9"	80		X	X		X	
MX-100 III	995	9"	80		X	X		X	
HR-1	1,295	15"	16	X		X		opt	
DP-9501	1,700	15"	200		X			X	
StarWriter	1,995	15"	45	X	X	X	X	opt	X
PrintMaster	2,495	15"	55	X	X	X	X	opt	X
Diablo 630	2,800	15"	45	X	opt	X	X	opt	X

Table 1.





Kenneth Bowles
The Prophet of Portability

by Allan Tommervik

Vivid images leap to mind when contemplating the man who is Dr. Kenneth Bowles.

First there's Don Quixote. Popular depictions show him as a silly old man, fruitlessly and rather ineptly attempting to tilt with a windmill. But *Man of La Mancha* presents another side of his character: sometimes the value of a quest is in the searching; sometimes it's necessary to dream the impossible dream. And if the world at large doesn't have the vision to see the same value, it's the world that is the loser.

Also conjured up is another Broadway character—Professor Harold Hill, title character of *The Music Man*. Hill sells band uniforms and musical instruments in hick towns in the Midwest. He trains his charges on imaginary instruments and skips town after collecting his money and before the townspeople discover the scam. But in *The Music Man* the damndest thing happens: when the band instruments arrive, Hill's pupils actually can play them.

Don Quixote Rides to Gary, Indiana. Someday in the not-too-distant future, schoolchildren will probably be taught the name of Ken Bowles as a major contributor to computer science as they'll know it then. Twenty years ago, that was a totally improbable event, and the odds weren't too high even ten years ago.

In 1972, Bowles was teaching in the computer science department at the University of California at San Diego and was director of the computer center for the university. That's not a scene exactly optimum as the springboard for making history.

First, San Diego is not the city whence one would suspect momentous things. The elders of that fair southern California metropolis are renowned for considering John Birch a left-wing agitator. Anything newer than an Edsel is looked upon with suspicion. Not everyone down there has been convinced that dial telephones aren't a Communist conspiracy. As for computers, their main virtue is that they can be unplugged.

Then there's the other half of the populace. The youth of San Diego aren't exactly exalted for their seriousness of purpose. That area of the California coastline is blessed with outstanding beaches, and the kids clearly believe that God wouldn't have put all that sun and sand there unless he meant it to be used. If fifties haircuts are the badge of admission to the elder set, suntans perform the same function for the teens.

The University of California at San Diego was not the school whence anyone would expect major contributions. The California university system is justly famed for the quality of its education; but much of that reputation was earned by the Berkeley campus. True, UCSD had higher standing than its sister school, San Diego State College. At least nobody accused it of offering Basket Weaving 256 as a graduate class.

In terms of the standing of their computer science department, the kindest comment was that the students learned something. Nobody uttered UCSD in the same breath with Stanford, Michigan, Dartmouth, and other computer science biggies. In fact, nobody uttered UCSD in the same breath with the words computer science.

So what you found in San Diego was one of the longest-shot parlays in history: solid citizens yearning for the good old days, kids who didn't care as long as the sun was out and the surf was up, an undistinguished computer science department, and a professor who was dividing his time between managing the school's computer resources and teaching.

Donkey Songs. It was in this environment that Ken Bowles and a bunch of present and former students produced Pascal on a p-System and proved portability as a viable concept. The p-System is the computer's equivalent of the Gatling gun or Henry Ford's assembly line. Just as the Gatling gun increased the firepower of the indi-

vidual soldier and just as Ford increased the productivity of his workers by introducing specialized tasks on an assembly line, the p-System provides the potential for increasing programmer productivity.

While it probably could have happened under another set of circumstances, it's not clear that there would have been a UCSD Pascal and a p-System if Bowles had not been relieved of his job as director of the university's computer center.

Freedom from the responsibility of overseeing UCSD's computer facility allowed him to address a pressing problem within the department: how to design a low-cost computer resource for students of computer science.

Bowles and others in the department had already made a contribution to mainframe computing by developing a job control system to work with the operating system of the Burroughs 6700. UCSD had purchased the 6700 because of its memory swapping capacity but had found the operating system somewhat finicky. Jobs had to be run in a certain order or the computer would declare an unscheduled lunch. This put significant responsibility on the operator to maintain the job queue in the proper order to prevent system crashes.

While Bowles was director of the computer center, a UCSD team developed a job control patch for the operating system that Burroughs subsequently adopted.

By 1974, it became clear to the faculty at UCSD that a low-cost computer resource was necessary for student programmers. Computer time to run student assignments was costing more than five dollars an hour. At that time, the computer science department was teaching Algol, the native tongue of the Burroughs mainframe. But they were casting eyes at the nascent microcomputer revolution and recognized that Algol was inappropriate for the smaller machines.

Big Band Blues. Nearly simultaneously, Pascal was attracting a lot of attention on these shores after its design in Europe. Bowles felt the language held great promise to achieve the portability he felt was necessary. But the design specifications he drew up would have daunted a crack team of advanced systems developers from IBM.

The foremost requirement, of course, was that the software have a sound instructional base. Bowles was hoping to maintain the instructional values of Algol, but this meant creating significant extensions to existing Pascal.

Almost of equal importance was that the software be portable. UCSD had gone through two changes of instructional hardware, and each had entailed several months and many man-years of effort in adapting the software to the new machine.

Finally, Bowles sought a design with enough built-in features that the system itself could be used to develop and maintain its own software.

Digital Equipment Company came to the university's aid with a PDP-11 to help with the development task, and Bowles and his students were off and running.

Grad student Mark Overgaard surfaced early in the project as the tech guru and participated in many of the nuts-and-bolts design decisions necessary to implement Bowles's vision. With total disregard for caution and playing it safe, the group wrote the compiler for the system in Pascal itself. Talk about bootstrapping! That's roughly akin to developing the principles of architecture while building the Taj Mahal.

It would have been a pip of an idea if the purpose had been to provide an interesting illustration of recursion in real life. But anyone using a developing language to implement a new system will never be accused of coppering his bets. It was a lot like *The Music Man*. Hill's bunch of kids shouldn't have been able to learn music on imaginary instruments, and Bowles's kids shouldn't have been able to program a system in a language they were extending. But when the sheet music and instruments arrived, the kids could play. And when Pascal was booted on the p-System, it ran.

One of the hallmarks of youth is that it doesn't recognize what can't be done. Many cynics have attributed similar shortcomings to academicians. But whether it was a bold stroke of genius or just plain old dumb luck, the fact remains that the design technique worked.

There's a lesson here for everyone. The UCSD group accomplished on a shoestring in about thirty months what the National Science Foundation and the Department of Defense spent millions of dollars and several years to do—the development of a portable system.

The final proof of the pudding came in late 1977 when all the elements of the p-System were successfully ported from an LSI-11 to an 8080-based machine made by Northwest Microsystems.

Until that time, Bowles confesses that neither he nor anyone else actually understood the power of the system they had built. But that demonstration of portability underscored its strength.

Sancho Panchez's Big Byte. It's here that the Bowles story takes an unlikely twist. Were Bowles the typical academician, he would

have called a bunch of publications with titles including words like *Journal of* or *Proceedings of* and offered learned papers on the successes occurring under his supervision. That's the typical method of announcing one's achievements to one's peers in the world of academia, and it probably would have been the death knell of the p-System. It's frightening how many good ideas get buried in such learned tomes.

To Bowles's eternal credit, he recognized the value of what he had wrought. So instead of turning to the educational media, he talked to *Byte* magazine. *Byte* published an article written by Bowles, and pandemonium broke loose.

The computer science department received more than one thousand inquiries about the system.

In short order, one section of the department became a cottage industry all its own. Paid students replied to inquiries and copied disks. Other student staffers helped draw up licensing agreements and participated in negotiations.

In the abstract, there would seem nothing more innocuous than a group of students showing an industry the error of its ways. And that's essentially what the UCSD kids were doing to the computer industry. But things always seem to be more complicated in these modern times, and controversy was just around the corner. University management felt Uncle Sam's Internal Revenue Service might take strong exception to the goings on in San Diego.

The problem stemmed from a gentleman's agreement between the IRS and the University of California system that the system could file one consolidated tax return. Foundation of that agreement was that the system had substantially no income from operations other than education. Suddenly Bowles and company had changed that. They had prospects of filling the university's coffers with licensing lucre.

The university feared the wrath of the IRS. In this day and age, the wrath of the tax collector is at least as mighty as the wrath of the righteous, so change was clearly in order. What the IRS could require was separate tax returns for each campus in the system. Just the additional paperwork involved would make that a costly requirement for the Board of Regents, who opted for the status quo.

We Got Trouble. The university advised Bowles that there appeared to be three alternatives: the project could be closed down permanently; the project could be closed down temporarily while UCSD appealed to the IRS for an exemption from the ruling; or the project could be placed in the hands of a trustee company, which would remit royalty payments to the university.

The first two choices were anathemas to Bowles. One killed the project and the other was tantamount to killing it, because an exemption might have taken as long as eighteen months to procure. So Bowles went courting a suitable suitor.

The university required that the company should have a southern California base and be experienced and respected in the software business. Neither of the independent microcomputer systems houses then extant—Microsoft and Digital Research—had southern California offices, but big systems company SofTech, a Massachusetts firm, had a small branch office in Escondido. SofTech converted that branch into an independent subsidiary and struck a deal.

Even here, controversy arose. The original contract called for Bowles and Overgaard to spend 50 percent of their time at the university and 50 percent with SofTech for a period of one year. That proviso was more important to SofTech, because it ensured continuity of development during the changeover. In addition, Bowles was to have oversight responsibilities for the university.

But state of California conflict-of-interest laws forbid someone who has oversight obligations from receiving stipends from a company operating under the contract the person is overseeing. So at the last moment, Bowles was forced out of the ongoing effort.

Soon thereafter, Bowles lost his zeal for Pascal. By this time he had become a passionate proponent of standardization and portability, and he felt that some of the independent extensions of Pascal added by some of the licensees violated the fundamental precepts of standardization. He relates that "it was painful to sit through two hours of debate about whether to follow a command with a comma or a semicolon, when from an engineering standpoint it made no difference."

So now he calls himself "a fugitive from Pascal."

Where has he fled? To Ada, the language that will soon, by fiat, be the official programming language of the Department of Defense. Ada was designed by a team from Honeywell under government auspices.

Double Dulcinea. The implications are staggering. Ada is brought to you by the same folks that bring you price supports to encourage tobacco production while telling you that smoking may be hazardous to your health. Ada is brought to you by the same fine people who bring you the U.S. Postal Service.

Not only that. It's been defined and standardized for the primary purpose of discouraging impromptu improvements. As a non-changing language, it joins the ranks of Latin, a language which, you may have noticed, is rarely spoken in civilized conversation these days.

With these thoughts foremost, how bright is it to embrace Ada?

It's very bright, if you're a man of Ken Bowles's persuasion. Bowles speaks eloquently of programmer productivity and the enhancements potential thereto in a truly standardized language.

Bowles weaves dream castles of software "chips," pieces of software analogous to hardware chips. In standardized systems, the user will be able to select various configurations of software chips for particular applications, just as various hardware chips are chosen for similar reasons when configuring a machine.

The future, he believes, will belong to software, with the application being the raison d'etre for purchase and the hardware being chosen for its compatibility with the application software. In that future market, a standardized language will stand software developers in good stead.

On answering queries about whether even the government can enforce standardization of a language in use, Bowles tends to temporize by telling of various seminars and conferences where that question has been raised. Each time, the moderator has phrased the question to the audience as, "Do you hope the government will stand firm on a standardized Ada?" When the question is phrased that way, Bowles reports, the vast majority respond in favor of standardization.

After telling of those experiences, he'll address the question directly by saying he hopes, and he believes, that the Department of Defense has the will and the clout to make standardization stick.

San Diego Charged. Bowles is putting his money where his mouth is. He's formed TeleSoft, now a forty-person company engaged in bringing Ada to market. He serves TeleSoft as chairman of the board while still serving as a faculty member at UCSD on a half-time basis.

TeleSoft has subsets of Ada running on DEC and IBM mainframes as well as on the IBM Personal Computer. Bowles hastens to inform that it's legitimate to provide subsets of Ada only so long as the commitment remains firm to provide all the extensions of the language as soon as practicable. As he eulogizes Ada, much of a listener's cynicism toward the language subsides.

After all, eight years ago, if Bowles had explained his vision of Pascal to you, you would have perceived him as Don Quixote, out there tilting at windmills. Instead, he was the Man of La Mancha, daring to dream the impossible dream and making it come true.

Ken Bowles changed the realities of languages and operating systems, and computing will never be the same.

So Ada has a powerful ally when Bowles comes down on its side. In fact, with Bowles in its corner, Ada may be strong enough to withstand government support. ▲

PERSONAL COLOR GRAPHICS

The PrintaColor PG-1000

Personal Color Graphics Has Arrived
Turn your IBM PC, Apple II, or Apple III into a color graphics system by adding a PrintaColor *PG-1000.* Personal graphics ...in color...at your finger tips. The *PG-1000* transforms dull and difficult to understand graphics into easily understood vivid color images. PrintaColor's *PG-1000* is the perfect blend of printer capabilities, making available for the first time one printer that combines quality text, graphics and color.

Quality Performance
The *PG-1000* is designed for the professional who requires high quality color output in minutes, not hours. Utilizing ink-jet technology, precision images containing over 125 shades of colors can be produced. An 8½ by 11 inch image is produced typically in under two minutes. 1440 dots per line give your copies the crisp definition required for detailed graphics.

Easy To Use
PrintaColor's *PG-1000* is designed to be at home in any office environment. Whisper quiet operation and compact size allow the *PG-1000* to be your desk-top printing companion. Color images are produced on readily available computer paper. The ink is contained in a snap-in, disposable cartridge which inserts in seconds, producing on average over five hundred color copies. There is no waiting for inks to dry, pens to clog, and no ribbon to replace. Multiple copies are as easy as keying in the number desired...each an original.

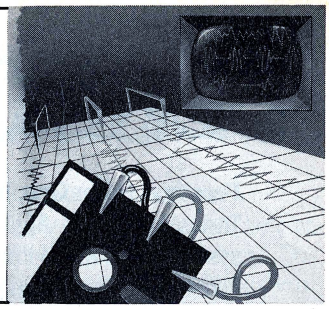
A Complete Printing System
The *PG-1000* is a complete color printing system including everything you need from the printer to a box of paper. From the moment you unpack your printing system you are ready to begin producing color graphic images with your personal computer. An easy to install interface card and cable connects the printer to your system. Software is included to facilitate copying the graphic screen in color, whether you use a black and white or a color monitor. A box of paper and an ink cartridge rounds out a complete compliment of items that allow color printing from your own graphics software or many commercially available software packages. Retail price $4,995.

Built For Reliability
The *PG-1000* utilizes a rugged carriage designed for long life. It has a rated life greater than 6000 hours MTBF. The color head has a proven reliability of greater than 10^{10} operations. A six month warranty is standard with factory maintenance contracts available on an annual basis.

For the name of your nearest dealer call or write

PrintaColor
CORPORATION
5965 Peachtree Corners East
Norcross, Georgia 30071
(404) 448-2675

marketalk reviews



Unless otherwise indicated, software reviewed runs in DOS on machines with either display adapter and requires 64K and at least one disk drive.

EasyWriter Version 1.1. This is a revision of the *EasyWriter* originally released by IBM. The new version has been extensively reworked, and most of the bugs in the original have been fixed. Nevertheless, the program retains the flavor of the earlier version.

Some features will seem familiar even to those who used *EasyWriter* on the Apple. From the file system menu, you hit G to get a file, E to edit, S to save, R to revise. But there are surprises as well. You can hit F4 to go to the additional commands menu if you want, for example, to set tabs or align text. Conveniently, the additional commands menu can be reached with a single keystroke from either the filing system or the editor.

EasyWriter 1.1 uses standard DOS files. The long, continuous files can be troublesome on occasion. For example, when you align text the program automatically aligns everything from your cursor to the end of the file. That can take a while. On the other hand, the standard files are easy to transfer, and the program can be used to edit programs or *VisiCalc* files.

All in all, *EasyWriter* does live up to its name. The manual is a tutorial that leads you through the straightforward steps of editing and printing. (Of course, a tutorial is not very convenient for quick reference, but the index helps make up for that.) The program makes full use of the IBM keyboard, including the arrow and function keys, though quite a few control key commands remain. It is possible for an experienced user to pick up *EasyWriter* and be off and typing in an hour.

There are a number of useful advanced features—search and replace, a word count command, an undelete key. Lines can be set centered or justified. You can also "print to screen" to see how your text will look on the printer.

However, a number of ugly features remain, some inherited from the earlier versions. *EasyWriter 1.1* relies too heavily on embedded commands for formatting. To underline a single word, you have to insert four lines in the text, including two embedded commands. Even worse, the program is not intelligently organized. The commands are jumbled—file menu for this, help menu for that, control key for this, function key for that. Sometimes it can be very confusing. Right margins are set in the additional commands menu, but left margins are set by embedded commands.

EasyWriter 1.1 is neither the cheapest word processor on the market nor the best, but it is certainly worth the price. **fj**

EasyWriter Version 1.1, IBM (Box 1328, Boca Raton, FL 33432; 305-998-2000). $175.

Macro Assembler. The *Macro Assembler* from IBM is a powerful implementation of 8086 assembly language. This version of the Intel Asm 86 language supports all the data pseudo ops—including *structure* and *record*, which aren't supported even on some assemblers that run on minicomputers. It also handles most of the attribute operators, such as *ptr* and *this*.

The *Macro Assembler* language is essentially composed of two parts: the pseudo op-codes and the instruction op-codes. The pseudo op-codes are further broken down into four types: data, conditional assembly, macro, and listing codes.

There are twenty-five pseudo ops in all. Among them are the *segment-ends* and *proc-endp* directives, which enable the programmer to develop highly structured source code in modular form. Modules so developed may be linked together after they're assembled, which makes for easier program development. Segments in one module can also be declared as *public*, which concatenates them with segments of the same name in other modules. At link time, all of the intermodule references are cleared up.

The *structure* pseudo op is probably the most useful one in the language. With this command you can set up arrays of data, with each array containing several fields and each field having a label. The labels then provide access to the data.

Another interesting data pseudo op is *include*, which, when encountered by the assembler, causes the file named in the *include* statement to be read from disk and incorporated into the source code of the file being assembled.

There are twelve conditional assembly pseudo ops. These statements are generally of the form: *if [argument] . . . (then) . . . else*, where *then* is implied. Conditional statements allow different portions of source code to be assembled, depending on the truth status of the argument.

The macro commands are a group of eight pseudo ops that are used to define and name recurrent sections of code. Instead of having to type the code every time it comes up, the programmer simply types in the name of the macro. At assembly time the assembler substitutes the appropriate code for the macros. The macro facility allows the programmer to assign parameters to macros when they're called; these parameters may change the definition of the macro. Also included with the macro commands is a repeat function that operates similar to a for-next loop in Basic. The macro facility is clearly the most powerful feature of the assembler package, and the descriptions and examples in the manual will help programmers make maximum use of it.

The *Macro Assembler*'s 375-page reference manual is not written as a tutorial, and anyone thinking they can learn assembly language from it will be in for a surprise. As a reference manual, however, it is thorough and well organized.

The manual devotes 175 pages to the instruction op-codes and describes each in great detail. The format and purpose of each instruction are given, along with examples of usage. Better than simple format statements, the examples offer the programmer greater insight into the many addressing modes offered by the assembler. These addressing modes include immediate, indexed, indexed plus displacement, base plus index, and base plus index plus displacement.

The instruction set includes string handling commands that simplify such common operations as the parsing or comparing of strings. The string instructions can be prefixed with a repeat command to simplify the coding further.

The last chapter of the manual leads the programmer through assembly and linkage and execution of a sample program. By going through the sample session, the user can get a feeling of what it's like to interact with the assembler and linker. The sample program does a good job of showing how different commands and pseudo

ops are used.

One point that should be made about the object code files generated by the assembler is that they're in an encoded form usable only by the linker program. This file structure makes the *Macro Assembler* useless for someone who wants to download object code to another 8088-based computer.

There are 117 error messages listed in the manual. These messages make for easier debugging of the source code, since they are quite specific. Many assemblers simply flag errors as either addressing or syntax errors and leave it up to the programmer to try and locate the source of the problem.

The actual assembly process of the *Macro Assembler* is somewhat slow, since the program has to make frequent trips to disk. An assembly that generates approximately 2.5K of object code takes almost three minutes.

One of the major drawbacks of the *Macro Assembler* is not the fault of the program itself but the fact that it comes without an editor. The programmer who buys the assembler package had better invest in some sort of word processing program with which to write the source code, because Edlin (supplied on the DOS disk) is slow and clumsy. Had IBM supplied a cursor-based editor with the *Macro Assembler* package, the combination would have been unbeatable. All in all, however, the IBM *Macro Assembler* is a superbly documented, smooth-running program that enables the programmer to use the full power of the Personal Computer. **bp**

Macro Assembler, IBM (Box 1328, Boca Raton, FL 33432; 305-998-2000). $100.

1st Class Mail. By Bob Schoenburg and Steve Pollack. This is the IBM version of the highly regarded Apple II mailing list program from Continental Software. The authors have done an excellent job in transferring it to the IBM.

The features of *1st Class Mail* are indeed impressive. It is really more than a mailing list program. In fact, it's also useful as a database program. A dozen user-definable fields are allowed. And, unlike many other data management application programs, *1st Class Mail* lets you use an unlimited number of disks for a single mailing list. Given that multidisk capability, it isn't possible to speak in the traditional terms of how many addresses the program can hold. The answer is as many as you want.

1st Class Mail is not only useful for huge lists but also lends itself well to categorical disk filing systems. For example, disks can be categorized alphabetically or by states, sales regions, and so on. Files from various disks can be sorted and transferred onto other disks. This allows extracting special-purpose mailing lists from more general ones. For example, regional mailing lists can be created for use by individual offices simply by extraction, keying on ZIP code or city-county-state fields.

The program is menu driven, and the options on the menu are quite clear and easy to follow. The menus are coordinated with the documentation and lead you handily through list maintenance.

The instructions do a good job of helping you understand how the program works. There is no index but there is a table of contents. The first text you encounter is an overview of the system. Hardware limitations, data capacities, and software capabilities are generally discussed. Many of the outstanding features of the program are discussed there. A section is included that tells you how to use the user guide. Many hints and suggestions are included. The very first page of actual instructions clearly outlines the dos and don'ts of program use. All in all, the first steps in the manual are conducive to a successful first encounter with the program.

The documentation is cleverly keyed to the menu displays. There are no photographs (none are needed), but the page layouts are very informative. Left pages show menu displays and the right pages explain menu options in detail. Often you will be referred to other pages in the book by goto statements. This nesting is complete and mimics the actual software menu nesting. The subsections even tell you which page to return to. This is a very well considered, people-oriented documentation package.

For comprehensiveness, ease of use, flexibility, and productivity, *1st Class Mail* can be recommended without reservation. **bt**

1st Class Mail, by Bob Schoenburg and Steve Pollack, Continental Software (11223 Hindry, Los Angeles, CA 90045; 213-417-8031). $74.95.

Space Guardian. This is a version of the venerable *Star Trek* game, which has been developed for many computers over the years. Graphics are limited to standard text characters, which means the game can be played with either a color or monochrome adapter.

The universe is an eight-by-eight grid. Each of the sixty-four "quadrants" is further divided into an eight-by-eight grid. In each quadrant are a number of stars whose main function is to obstruct your movement. There may also be a number of hostile aliens and possibly a base where you can refuel. Your ship does eat up fuel. Shields, phasers, and engines all take fuel, and it is very difficult to finish the game without refueling at least once.

You move your ship by specifying distance and direction. Both are very flexible; it is no trick to move 3.72 light-years on a course of 163.5 degrees. Combat is even simpler; you just allocate energy points to shields and phasers. Phasers always hit but do not always destroy. The alternative torpedoes do not always hit, but always destroy when they do. Since aliens are distributed randomly, you can end up fighting several in the same quadrant. Aliens may fire more than once and may get the first shot, depending on the level of difficulty. If nothing else seems to work, you can go into hyperspace, which teleports you to a random quadrant.

Anytime you enter combat, you are likely to take damage. A shield hit usually just subtracts from the energy available. On the other hand, a weapons hit may prevent you from firing back. Damage to short-range sensors prevents you from viewing the quadrant. Damage to long-range sensors prevents you from seeing the adjacent quadrants. Damage to the warp engines cuts your maximum speed to 0.2 and prevents you from entering hyperspace. Your ship can lose several or even all of these systems at the same time. Damage is repaired gradually, based on how many light-years your ship has traveled. It can also be repaired when you dock at a base.

This game has many attractive features. The fact that you don't know what's in a quadrant until you see it on your sensors adds an element of exploration. The inevitable ship damage challenges your ingenuity. The time limit (measured by light-years traveled) gives the game some tension.

On the other hand, the game prickles you with small irritations. The keyboard buffer is disabled; you are perpetually waiting for prompts. Every time your ship moves, you have a random chance of taking damage from unknown sources. The light-years traveled, as recorded by the computer, do not always match dead reckoning. The random game setup may make it impossible to win at all.

Space Guardian does show some improvements over the earlier games of its genre. It is worth buying if you prefer strategy games to shoot-'em-ups. **fj**

Space Guardian, Omric (1268 Main Street, Newington, CT 06111; 203-666-4240). $29.95.

VisiDex. By Peter Jennings. *VisiDex* is an informal database and agenda manager designed as a cross-referencing electronic card file. The unit of storage—the equivalent of a record on more formal database systems—is a screenful, or portion thereof, structured any way you please. The program presents you with an essentially empty screen, rather like a blank 4x5 index card. You put whatever information you need to record on it, then designate up to thirty keywords—which may or may not be part of the contents of the screen—and send the card off to disk. The card can subsequently be retrieved by any of those keywords.

The system is ideal for writers or researchers who wish they had some magic way to cross-reference their notes. It's also handy for desk workers who jot a lot of notes to themselves. *VisiDex* cards may be stored under keydates as well as keywords. If you have an appointment on the seventeenth of September, you can sock it away

on disk and *VisiDex* will remind you when you boot the program that day. You can also request up to fifteen days of warning, so the program will flash your date screen in advance of the date in question. Other agenda options allow you to bring up reminders on, say, the third Friday of every month or on a specific date of every month—handy for that monthly poker game.

A third way to use *VisiDex* is to create template screens that have field names already in place, in inverse or flashing characters. When you're in the template mode, the program will bring up a fresh fielded screen each time you send data to disk. Single control keystrokes then take you from field to field. Template screens can be stored under both the template keyword and normal keywords.

Searching in *VisiDex* is done either by keyword or by text. The former is quicker, naturally, since the program first consults an index and then goes directly to the appropriate disk sector. But the text search is handy, too; it allows you, among other things, to specify comparative operators in your search string, so that you can, for example, track down all donors who have contributed over a certain amount or all clients living within a particular ZIP range.

Print options include reports by individual keyword or sorted within a certain alphabetic range of keywords (all keywords starting with the letters F through Q, for example). Screens that have the same keyword appear in printouts in the order in which they were filed. Another option allows you to print only a portion of each screen, so, for example, you can keep mailing list entries in the top six lines of each screen, with comments below, and then suppress the comments when printing.

The manual is solidly built and well written. It's divided into tutorial and reference, with a handy tack-up command chart in the middle. You may never need the chart, because all available command options in each major division of the program are displayed at the top of the screen. Hitting the space bar within a command brings up additional help.

Along with all this power and flexibility, there are some limitations. Each data disk you create is its own world with its own keyword index. You can't chain one disk to another. You can, however, transfer screens between disks. The program is designed for single-sided floppies and allows up to 2,000 keywords per disk.

The biggest limitation has to do with speed. For some reason this incarnation of *VisiDex* is maddeningly slow. Where the Apple version would fetch a screen in about no time at all, the IBM model may take ten or fifteen seconds. This version has to get a program overlay from disk each time you call a new subdivision of the command structure. Single-drive users should check it out in the store to be sure they can live with this much swapping. Finally, beware of the disk initializing command. Somebody forgot to put in a backout option there, so if you type /SI by mistake, you've no recourse but to shut off the machine and start over.

VisiDex is copy-protected. The publisher will sell you a backup for $20. **CS**

VisiDex, by Peter Jennings, VisiCorp (2895 Zanker Road, San Jose, CA 95134; 408-946-9000). $250.

The Speed Reader. By Richard Eckert, Jr., and Janice G. Davidson. Who says you can't get exercise sitting in front of your computer? Now you can do your daily ocular calisthenics with the help of *The Speed Reader*. Actually this program does more than stimulate your eye muscles. It provides a regimen of exercises and some clear instruction about how to form and maintain good reading habits.

The first two of the five exercises are warmups to train your peripheral vision. In the first, the computer displays a square in the center of the screen and then flashes a pair of letters to either side. The idea is for you to focus on the square and then perceive the letters without moving your eyes. Fifteen speed levels make your work progressively more difficult; when you succeed on one level the program bumps you up a notch, until you get to level fifteen, whereupon you can repeat the exercise taking in three letters at a time—and then four. The second warmup exercise does essentially the same thing with words. Here you have more data to take in in the same amount of time, but you begin to see the extent to which meaning assists perception: "lawyers jog cautiously" may be easier to apprehend in a split second than R P Q.

The eye movement lesson flashes a reading passage on the screen one phrase at a time. The phrases appear alternately on the left and right sides of the screen, so you're forced to give those eye muscles a workout; the idea is not to use your neck. You control the speed of the flash.

The column reading lesson gives you a reading passage in a narrow enough format that you should be able to take it in without lateral eye movement. As with the warmup exercises, the purpose here is to encourage you to use peripheral vision. And finally, there's a reading passage lesson that more closely simulates the appearance of a printed page. You set the speed in this one in words per minute; the program allows a range of 200 to 2,000 wpm.

For the three lessons—eye movement, column reading, and reading passage—the program uses a separate data disk of short reading selections. There's a quick multiple-choice comprehension quiz at the end of each. In addition to the two disks, *The Speed Reader* comes with a lucid forty-four page booklet that teaches the elements of good reading technique and suggests a daily workout plan to follow in doing the exercises. **CS**

The Speed Reader, by Richard Eckert, Jr., and Janice G. Davidson, Davidson and Associates (6069 Groveoak Place, Rancho Palos Verdes, CA 90274; 213-378-7826). $74.95.

Astro-Dodge. By Dave Gangola. This program will be familiar to players of the arcade game *Asteroids*. Once again, you are a starship captain who has gone mad and decided to destroy an asteroid field with shipboard weapons. (Your crew really ought to mutiny!)

The asteroids are well drawn. They drift menacingly across the screen at varying speeds. When you destroy one, it breaks into smaller parts that must be destroyed in turn, producing (you guessed it) even smaller pieces of rock. Touching any one of these

PC/FORTH

Complete FORTH program development systems for the IBM® Personal Computer. Packages include interpreter/compiler with virtual memory management, line editor, custom screen editor, assembler, decompiler, utilities, file and record access modules, and many demonstration programs. 100 page user manual $100.00

Floating point arithmetic extensions and cross compiler available at additional cost.

Ready to run under PC-DOS or CP/M-86®. Standalone version available soon. 64 kbytes RAM and two disk drives recommended. (IBM is a registered trademark of International Business Machines Corp. CP/M is a registered trademark of Digital Research, Inc.)

Laboratory Microsystems
4147 Beethoven Street
Los Angeles, CA 90066
(213) 306-7412

prolific asteroids causes you to lose your ship in a pretty explosion. As if that weren't enough, there are a couple of alien ships that keep using you for target practice.

Your ship has no armor or shields; there is nothing between you and death but your blasters and your ability to maneuver. Unfortunately, the program can't keep track of more than four blaster bolts at one time. That means your guns may jam at a very awkward moment. Movement is more or less in accordance with the laws of physics—acceleration is gradual, even when you would like it to be instantaneous. If all else fails, you can hit the space bar and disappear into hyperspace. (Perhaps to reappear inside an asteroid!)

If you destroy all the asteroids, you are rewarded by a fresh screen with even more asteroids. If you lose a ship, it is replaced automatically. This can happen twice—more if you are racking up the points. A replacement never pops up inside an asteroid, but it may appear directly in the path of an alien blaster bolt, which makes for a short game. If you manage to overcome these difficulties, you can try playing the game at a second level of difficulty.

In general, the graphics show expertise, but the programmer managed to leave a blank area off the bottom of the screen. If a ship or asteroid moves into that area, it effectively becomes invisible, and, since aliens tend to make their appearance in the blank area, this puts the player at a serious disadvantage. To complicate matters, it is not always clear which direction your triangle-shaped ship is facing. Also, the ship can face in only twelve directions, so it is sometimes impossible to shoot an asteroid that approaches from an odd angle.

Nevertheless, this is a fast-paced, addictive game. If you have the required color adapter, give it a try. **fj**

Astro-Dodge, by Dave Gangola, Digital Marketing (2670 Cherry Lane, Walnut Creek, CA 94596; 415-938-2880). $39.95.

The Home Accountant Plus. By Mike Farmer, Robert Schoenburg, Stephen Pollack, and Larry Grodin. This is a translation of a program that's been enjoying considerable success in the Apple market. The *Home Accountant* is a single-entry accounting system designed for individual, family, or small business use. It handles a maximum of five checking accounts, five cash accounts, and one hundred budget categories. The program provides for chaining of data disks, so there's no apparent limit to the number of transactions that may be recorded.

Three modes of transaction data entry are available: checking, credit cards, and cash. The program treats accounts and budget categories, in most respects, as functionally equivalent, so that, for example, to transfer funds from your checking account to your cash account you just write a check and indicate "cash account" as the budget category.

Budget categories are of five types: asset, liability, credit cards, income, and expense; inclusion of the first two allows the program to generate a comprehensive personal financial statement.

Seven fields are provided for data entry: date, payee/source, amount, memorandum, budget category, tax status, and a numeric code. The code is used for check or deposit number in the checking mode but can be used for whatever you please elsewhere. The checking portion of the program also tracks the reconciliation status of your transactions.

The *Home Accountant* stores transaction information in one large file on disk but requires you to open and close months as you go nevertheless. The monthly partitioning of the data simplifies search procedures; it also allows the program to record preestablished automatic transactions whenever you open a new month (these are always dated as though they occurred on the first of the month). If you forget to enter a transaction before closing a month you are allowed to record it anyway, but it will show up in the wrong month when you later search for it.

Searching may be done on any of the data fields, singly or in combination, within any specified range of months. Each transaction meeting the search criteria appears by itself on-screen and may be sent to the printer, but there's no facility for an on-screen summary of selected transactions.

You can, however, get a totaled printout of all transactions meeting selected criteria. The program is currently compatible with close to a dozen printers, including the IBM/Epson family, TI 810, IDS 125 and 560, Anadex 9501, NEC 8023, and Okidata Microline 80. If you have the IBM printer or its Epson equivalent, the *Home Accountant* will automatically send out the appropriate control characters to give you 132-column printouts on 80-column paper.

Other print options, besides exhaustive and selective transaction summaries, include the personal financial statement and various kinds of budget reports. For on-screen budget analysis, the program offers a couple of bar graph routines—one for the color adapter and one for mono.

The *Plus* in the *Home Accountant Plus* is a forecast module that lets you measure the ravages of inflation. For example, you can give the program a financial goal, a period of time, a hypothetical inflation rate, and a current rate of investment return, and it will tell you how much to set aside each month.

The major strength of the *Home Accountant* seems to be its ability to handle a large number of accounts and transactions interactively, so that you never record a transaction more than once. The program is also easy to learn and use; the one hundred page manual is thorough and well written and does a decent job of hand-holding through the startup procedures.

On the negative side of the ledger, the *Home Accountant* is pretty much a literal translation of its Apple forebear and doesn't use any of the IBM's enhanced capability. Video display, for example, is constrained to forty columns, and the function keys are unemployed. This is less a criticism—it was a good program on the Apple and it's just as good on the IBM—than an indication of vulnerability to potential competition. **CS**

The Home Accountant Plus, by Mike Farmer, Robert Schoenburg, Stephen Pollack, and Larry Grodin, Continental Software (11223 South Hindry Avenue, Los Angeles, CA 90045; 213-417-8031). $150.

ELECTRONIC DISK **$69.95**
Will create the equivalent of a disk drive in your extra memory. Simply say how many 64K byte blocks are to be used by PC-DOS 1.10, and the rest of memory will be allocated to your fast electronic disk. Bonus: you can also allocate memory to a printer spooler! Really increases system throughput.

ISAM DATABASE **$69.95**
GET and PUT records to disk files by "KEY". Under 2 seconds.
Browse forward or backward in key sequence.
Update any part of the record, including the key.
Automatic recovery of disk space occupied by deleted records.
Keys do not have to be unique and can be any length.
Routines accessed from any Basic application. 6K overhead.
FAST! Sorts 5000 records in 12 seconds.
Includes mailing list application program & documentation.

STRING SORT **$24.95**
This machine language Shell sort routine will sort any string array into ascending alphabetical order. We believe it is the fastest sort available. Sorts 1000 strings in 2 seconds, and 10,000 strings in 30 seconds. Called from Basic:
BLOAD "SHELLSRT",0 : CALL SORT(N%,ARRAY$(1))

CHARACTER GENERATOR **$24.95**

CHURCH MEMBERSHIP **$69.95**

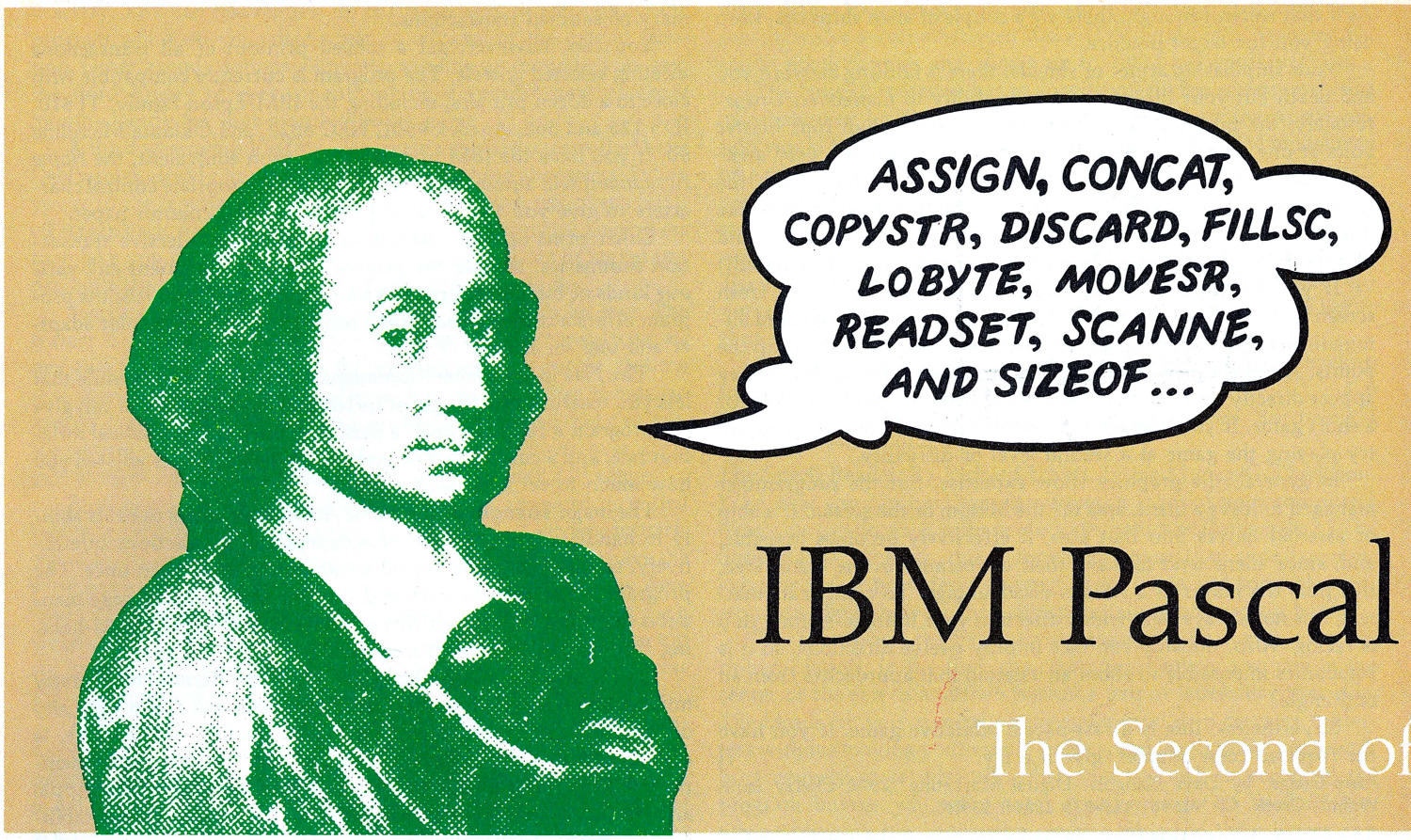
BLACKJACK,OTHELLO+8 *OTHERS ALL FOR* **$29.95**

WRITE FOR FREE CATALOG DEALER INQUIRIES INVITED

ENSIGN SOFTWARE
2312 N. COLE RD., SUITE E
BOISE, IDAHO 83704 U.S.A.

Telephone Order Line
For Bank Card Sales
(208) 378-8086

VISA MasterCard



IBM Pascal

The Second of

by Bruce F. Webster and Deirdre A. Wendt

The first of these articles (July 1982) discussed the various language features of Pascal and how the two Personal Computer versions (IBM and UCSD) have implemented them. Generally speaking, we found that IBM Pascal had more extensions to the standard Pascal definition than did UCSD Pascal, although UCSD had more predefined procedures and functions.

We also registered a certain wariness about IBM's complexity. Niklaus Wirth gave Pascal a clean, simple design to help make programs clear and predictable. The IBM version has lost some of that virtue and, as a result, is a little less predictable than the UCSD implementation.

Having covered language features, we're now going to tackle system features. This will include special execution modes, compilation, libraries, utilities, documentation, and other features.

A Tale of Two Compilers. A compiler translates the program that a user writes into code that the computer can execute, and the IBM and UCSD Pascal compilers differ dramatically. The IBM compiler requires three passes and produces 8086 machine code. The UCSD compiler requires only one pass (with occasional need for a linker) and produces a "pseudo-machine" code that requires an interpreter to execute.

Here's a description of what both compilers do.

IBM Compiler. The IBM compiler goes through three different steps—two compiler passes and a linking phase—to produce an executable code file. The first compiler pass produces two files. One is a symbol table—a list of identifiers with information for each. The other contains the program in an intermediate form. The second compiler pass uses these two files to generate an 8086 machine code version of the program. Finally, the linker merges this machine code with the necessary routines to produce a file that can be loaded and run.

IBM Pascal will compile three different types of program structures (IBM calls them "compilands"). The first is a complete program. The other two, modules and units, allow procedures and functions to be compiled separately and then used by different programs.

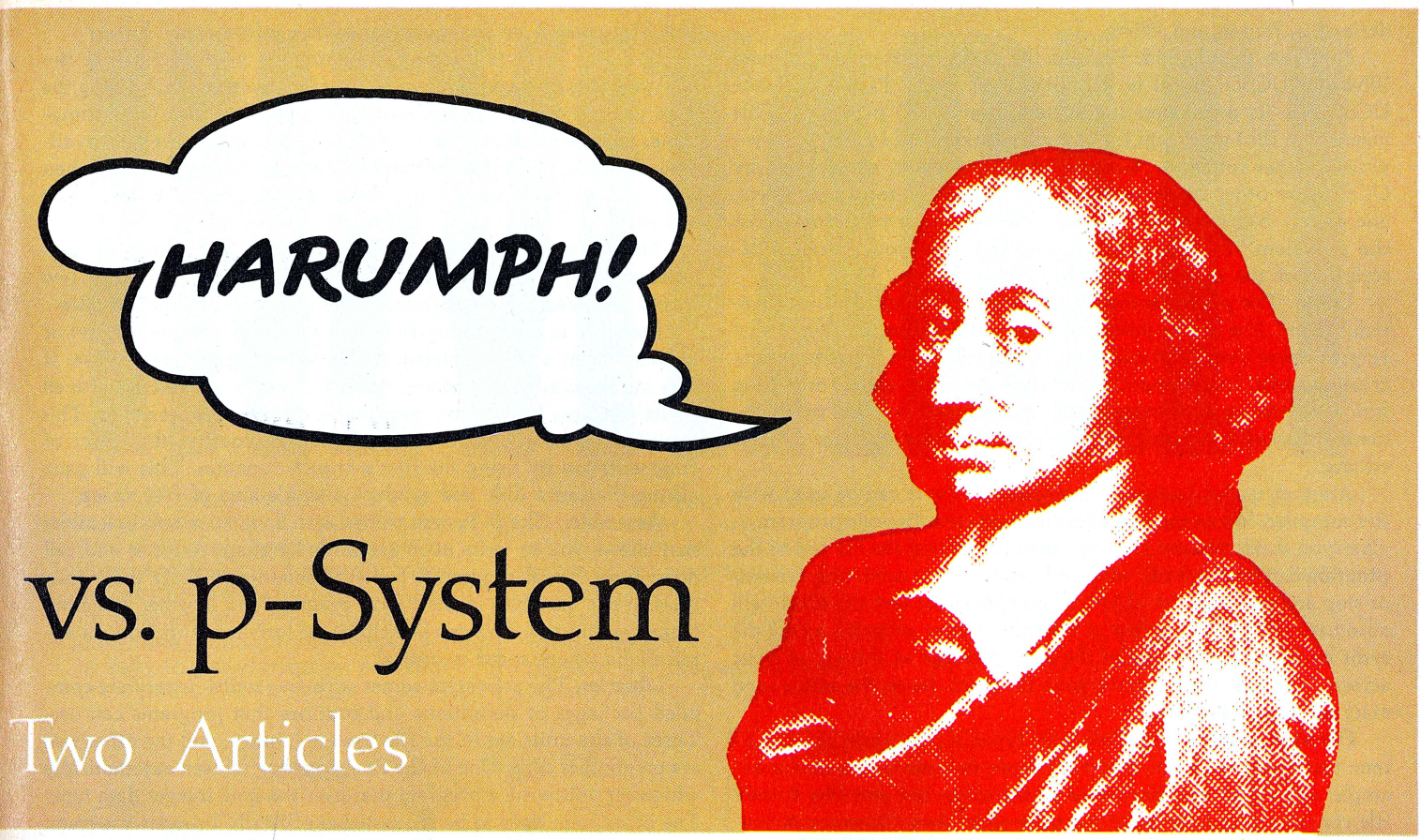
A *module* is simply a program without a body; it contains procedures, functions, and data structures (used internally only), but no main program body. A program using subroutines in a module declares them *extern*; the actual code is then linked in by the linker.

A *unit* is a package of data structures and routines divided into two sections. The *unit interface* contains the declarations that a program using the unit "sees." The *unit implementation* has the actual bodies of the routines declared in the interface. The unit implementation may also contain data structures and routines that are hidden from the user program.

A few last notes on the IBM compiler. The compiler requires 128K of RAM, so you can't do IBM Pascal development on a 64K system. Unfortunately, however, a finished program can't use more than 64K, even if you want it to. Since there are no (apparent) provisions for swapping code in and out of memory from disk, this 64K limit is a definite constraint on program size. Furthermore, the directive *packed* on data structures doesn't really cause bit packing, so there is no memory saving to be attained thereby. Finally, IBM uses its own set of compiler codes; it does not conform to the standards described in Jensen and Wirth.

UCSD Compiler. The UCSD Pascal compiler requires one pass (linking is only needed in certain situations), produces pseudo-machine code (p-code), needs only 64K, and generates programs that can use up to 128K.

The key to the UCSD system (known as the p-System) is the concept of the pseudo-machine, or *p-machine*. This is an imag-



vs. p-System

Two Articles

inary, stack-oriented microprocessor with its own instruction set (p-code). UCSD programs compile down into p-code; this code is then executed by a p-code interpreter written in the host machine's native code (8086 code for the IBM).

Input and output are handled through a basic input/output system (BIOS), also written in native machine code. Almost all the other programs in the UCSD operating system are in p-code. This is why the p-System exists for so many different processors and why UCSD programs are usually so portable.

One drawback of p-code is that it is almost always slower than equivalent 8086 machine code (although it takes up much less space). UCSD gives you two ways to try to overcome that loss of speed.

First, you can use the Codegen utility on your p-code files. This utility looks for sections of p-code that it can convert into machine code. However, Codegen will not necessarily convert all the p-code to machine language, since some p-instructions are too complex to be converted easily. Second, you can write 8086 assembly language subroutines, which your Pascal programs can then call.

The UCSD compiler, like the IBM, also provides for separate compilation of routines. First, UCSD supports *units*, just as IBM does (indeed, UCSD originated the concept of units). Second, 8086 assembly language routines can be declared *extern* and can be assembled separately. Finally, segment procedures can be compiled separately from other segment procedures.

The UCSD system facilitates the creation of large programs in three ways. First, you can set up the operating system to allow programs running under it to use up to 128K of memory. Second, you can use the *packed* directive to do bit packing. For example, if we make the declaration

```
VAR
    BITS: ARRAY[1..512] OF BOOLEAN;
```

then the program will set aside 512 words (1,024 bytes) of memory for the array bits. However, if we change the declaration to

```
VAR
    BITS: PACKED ARRAY [1..512] OF BOOLEAN;
```

then bits will only use thirty-two words (sixty-four bytes)—or one-sixteenth the space. The tradeoff for memory conservation is slower access to individual array elements.

Third, you can divide your program into segment procedures. Like any other procedure, a segment procedure can pass variables and contain its own set of data structures, procedures, and functions. However, a segment procedure is not read into memory from disk until the program calls it, and the memory it uses is liberated when the program leaves the segment procedure.

In other words, you don't need to have your entire program in memory at once. Also, you can control which segments are in memory at any time, by using the *memlock* and *memswap* procedures. Since you have up to 255 segments and since the segments may be compiled independently of one another, the UCSD makes it possible for you to create an enormous program.

The p-System also provides a limited form of concurrent procedures—procedures that can be executed nonsequentially. These routines are known as *processes*. Processes are not truly concurrent; instead, a given process will run until done, or until it's interrupted or forced to wait for something. Routines for semaphore synchronization between processes are available, and there are limited interrupt-handling facilities. Multiple instances of a given process can be running concurrently; each is considered to be a separate entity.

The UCSD compiler error codes include the set of error codes

defined in Jensen and Wirth.

Not Two Ps in a Pod. Perhaps the greatest difference between IBM and UCSD Pascal is that IBM runs under PC-DOS, while UCSD has its own operating system, the UCSD p-System. This means that IBM gives you the compiler, linker, and system library—period. Other functions, such as file management, are handled by DOS or by other software you've purchased (editors, assemblers, and so on). Since you can't use any of these other programs under the p-System, the p-System supplies them for you. Here's a description of the various features of the p-System.

Editor. The p-System comes with a screen-oriented editor (System.Editor). It is not as powerful as, say, *WordStar*, but it is a vast improvement over *Edlin* and is more than adequate for all program-creation tasks. Its command structure is easy to master. Formatting functions and an auto-indent mode assist you in writing nicely indented Pascal programs, while definable margins help in text editing.

Another useful feature of the editor is that it can be used with the compiler. When the compiler finds an error in your program, it gives you three choices: continue the compilation (so the rest of the program can be checked); stop and return to the operating system; or stop and return to the editor. If you choose the last, the editor will automatically put the cursor at the spot in your program where the error occurred and will print the error message at the top of your screen. You can then correct your program, leave the editor, and start the compilation again.

File Management. There are three types of files under the p-System: text, code, and data. A text file is one that contains text. For example, any file created by means of the editor is a text file. A code file contains p-code or 8086 machine code. Any file you can execute is a code file. All other files are data files. These include non-text files created by programs for saving information.

You can perform various functions on your files using the p-System file manager, known as the filer (System.Filer). Using the filer, you can create, list, rename, move, and delete files. Other functions, such as defining the default disk/file prefix, are also available. The filer is invoked from the main operating system; when you're done with your file functions, you return to the main system.

The p-System disks are divided into 320 blocks (640 on two-sided drives), and each block contains 512 bytes. The first few blocks are used for the directory; the rest are available for files.

Each file is contiguous; that is, it occupies a sequential set of blocks. When a file is deleted, its blocks are again free for use. If files are repeatedly created and deleted, the disk can end up with all of its free area broken into many little chunks between files. This condition is called fragmentation. You can counter the effect of fragmentation by using the filer's *krunch* function. This will pack all the files on a disk and produce a single area of free space.

Assembler. The p-System comes with an 8086 macro assembler that allows you to write 8086 assembly language routines and call them from Pascal programs. System pseudo-ops allow these routines to use variables in the Pascal program and in other assembly language routines. This can be helpful for low-level functions or applications where speed is critical.

Libraries. The p-System comes with five *units*—separately compiled packages of definitions and routines that programs can use. Three of the units are in the file System.Library, and the other two are on another disk. One of the five, *Longops*, is used automatically whenever you write a program that uses the long integer data type. The other four have to be requested specifically by your program.

Here are their names and the things they help you with:

Introducing low cost, IBM compatible disk drives

Up to 2.6 megabytes of IBM formatted storage

Low cost internal/external drives
Plug the INTERFACE IBM compatible disk drives into your IBM cabinet or add on to your existing drive configuration. INTERFACE drives can be used externally or internally, in a variety of configurations. To give you increased storage capacity up to 2.6 megabytes.

Internal Drives
- *Single sided drive with 168K bytes IBM format $215
- *Double sided drive with 325K bytes IBM format $335
- **Quad drive with 650K bytes IBM format $450

External Drives: (Includes drive, power supply, cabinet and cable)
- *Single sided drive with 168K bytes IBM format $325 (Maximum two drives externally)
- *Double sided drive with 325K bytes IBM format $445 (Maximum two drives externally)
- **Quad drive with 650K bytes IBM format $560 (Maximum two drives externally)
- Quad software diskette $95

*Compatible with 1.1 DOS
**Requires Quad software diskette

Up to four drives
You can have as many as four drives operating with your IBM personal computer. Giving you an incredible 2.6 megabytes of formatted storage.

Eight color cabinets to choose from
INTERFACE gives you a choice of color co-ordinated cabinets to choose from for external drives. Colors include standard off-white, an IBM-like blue, lime green, dark green, bright orange, brilliant yellow, black, and chrome.

Unconditional warranty and service
90-day unconditional warranty plus service center for out-of-warranty service. For more information about the IBM compatible disk drives, call or write today:

I² INTERFACE, INC.
7630 Alabama Ave., Unit 3
Canoga Park, CA 91304
(213) 341-7914

IBM is a registered trademark of IBM Corporation.
MasterCard, VISA and COD orders accepted.
Dealer and quantity discounts available upon request.

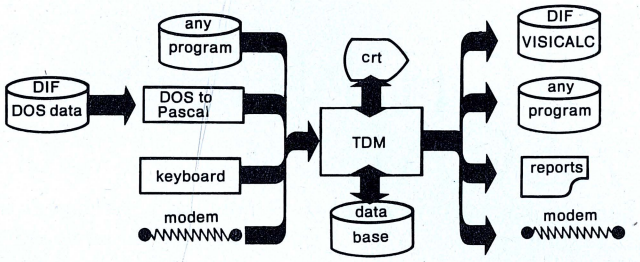
INEVITABLE!

As a serious computer user the biggest investment you will make in your data management system is time. And you can't afford to re-invest that time when your need for information expands. You need database management software that is easy to get started on and that anticipates the inevitable—growth.

TPG, The Program Generator, can be used to generate highly efficient user-structured inquiry, input, edit and file update applications programs without programming. In a program produced with TPG, up to 16 different TDM data files can be open and accessed simultaneously, with up to 17 different user-

As a business manager planning for growth your need for more and more information is inevitable. Your software should be chosen because it can handle the inevitable.

With the TDM/TPG Software System you will know you've done it right the first time.

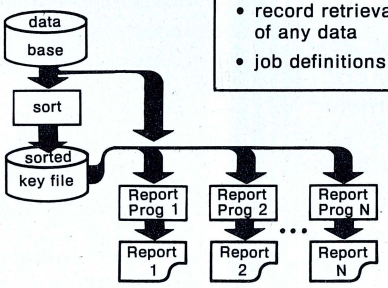


TDM can easily handle very large files, and includes the most versatile and powerful free-form report generator available for the IBM-PC, Apple II and Apple III. It provides simple reporting procedures in spite of its power and rapidly produces complex reports. When the inevitable happens and your requirements change, you can easily redefine and restructure an existing database.

With TDM you can combine any two databases to create a third. You can retrieve data and update files from remote locations. There are no rigid restrictions for report titles and headings. You can use GOSUB statements and labelled paragraphs. IF . . . THEN . . . ELSE statements can be nested 16 levels deep. TDM lets you sort *once* to produce any number of reports in the same sequence.

Other System Capabilities

- date arithmetic (for aging reports, elapsed time, etc.)
- interfaces with most text formatters and word processing programs to produce quality documents
- interfaces with Visicalc and Apple II DOS 3.3 files
- floating dollar signs, commas, leading or trailing signs, and negative amounts with brackets or CR
- 12 digit accuracy
- fast key sort, with keys built from portions or all of any fields
- interactive report processing, with rewrites to the file
- record retrieval and selection using combinations of any data
- job definitions



defined screen "forms." TPG makes it easy for you to change your custom programs as your needs expand. The TDM/TPG System can be used with most hard disks that support UCSD Pascal. On a hard disk, TPG can make it possible to access up to 500,000 records. It operates in both Corvus network environments. (Incidentally, there is no charge for additional copies for multiple-user installations.)

The TDM/TPG System is currently satisfying some very demanding managers at companies like Marriott's Great America, Saga Corporation, Fairchild, Intel, Corvus Systems, and Addison-Wesley.

TDM/TPG Hardware Requirements

Apple III or Apple II w/ 64K memory and 80 column board or terminal, 2 disk drives, or hard disk supporting UCSD Pascal.*

IBM-PC w/ 128K memory, 2 disk drives or Corvus hard disk.

*Certain restrictions may apply. Call for details.

Ask your dealer for a demonstration or call Pascal Systems at 415-321-0761 for more information. Pascal Systems, Inc., 830 Menlo Ave., Suite 109, Menlo Park, CA 94025

PASCAL SYSTEMS

IBMSpecial	Game control, music, some graphics, and keyboard
Turtlegraphics	Complete graphics functions
Screenops	Various CRT I/O functions
Commandio	Program chaining, I/O redirect

Utilities. The p-System comes with a significant number of utilities aimed at program development and system work. Among these are:

Setup	Allows user to change various system parameters.
Compressor	Converts one or more assembly language routines into a pure block of machine code.
Patch	Allows direct viewing and modification of disk blocks.
Decoder	Disassembles code files.
Debugger	Allows interactive debugging of programs.
Recover	Aids in salvaging files on disks with clobbered directories.
Tape	Transfers files between disks and cassette tapes.
Spooler	Prints out text files while you are working on the computer.
Codegen	Converts p-code to machine code.
Prnconfig	Allows you to configure your IBM printer (or MX-80) from the keyboard.

DATAMATE

...a *truly* all-purpose DATABASE MANAGER for **IBM PC** users.

A system that adapts easily to most any form including mailing list, inventory control and general information management.

- fewer management problems: multiple files per disk, no practical limitation on file content
- menu driven commands plus full use of **IBM** function keys
- cost effective: priced lower than comparable systems, free updates, adaptability
- custom applications available from author, assured compatability
- **DATAMATE...**TOTAL PACKAGE including manual, back-up disk, and sample data disk for $299.95.

Compu-Haus

1840 NW 50th Terr.
Kansas City, MO. 64151

TO ORDER, CALL 816-741-2669

Also available for the Osborne 1.

Portability. One key factor that should affect your choice of a Pascal is how and where you are going to use it and what other programs it will need to be compatible with. IBM Pascal runs under PC-DOS. This means that you should be able to run it on any IBM Personal Computer and that it should be able to use files created under PC-DOS. This may be a critical factor for you.

On the other hand, a UCSD Pascal program requires the p-System, and it cannot read or write files compatible with PC-DOS. This may be a serious drawback for certain applications. The offsetting advantage is that you should be able to compile your UCSD program on any computer—including Apple, TRS-80, Xerox, DEC, and others—that has the p-System version IV.0. The only awkwardness of compiling on another computer is that you may have to rewrite any IBM-specific code.

Documentation Dynamics. The difference in documentation between the two Pascals is dramatic. IBM Pascal comes with a single reference manual that's both hard to read and incomplete in its description of the language. Not only is it not for the novice user, it may be just as frustrating for the expert in need of detailed information. By contrast, UCSD Pascal comes with five binders containing six manuals. These documents tell you everything you'd want to know about the p-System (and then some), and they do it in a clear and well-organized manner. Here's a description of the manuals:

The *Operations Guide* is the smallest of the manuals; it gives an

PATCHING PASCAL

Some programmers have reported trouble with the modulo and multiplication operators in IBM Pascal. IBM has released a patch for the PAS2 disk, implemented as follows. First, make a copy of your PAS2 and put it in drive B. Put a DOS disk in drive A. Type in the following commands, which are italicized. (System prompts and responses are in nonitalic type. The letters XXXX represent memory addresses, which will vary.)

```
A > DEBUG
-LDS: 100 1 AF 1
-DDS: 175 L2
XXXX: 0175 22 02
-EDS: 175
XXXX: 0175 22.32 02.01
-DDS: 217 L4
XXXX: 0217 22-02 2B 00
-EDS: 217
XXXX: 0217 22.32
XXXX: 0218 02.01 2B.37 00.05
-DDS: 227 L4
XXXX: 0227 22-02 2B 00
-EDS: 227
XXXX: 0227 22.32
XXXX: 0228 02.01 2B.37 00.05
-DDS: 24D L4
XXXX: 024D 22 02 2B
XXXX: 0250 00
-EDS: 24D
XXXX: 024D 22.32 02.01 2B.37
XXXX: 0250 00.05
-DDS: 25D L4
XXXX: 025D 22 02 2B
```

overview of the system with emphasis on getting started. The *User's Guide* contains detailed descriptions of the editor, the filer, and the system utilities. The *Beginner's Guide* is a textbook for the complete novice. It serves as an introduction to the p-System and UCSD Pascal.

The *Pascal Reference Manual* is actually two manuals in one. The first gives the language definition and description. The second has thirty example programs that show how to use different features of the language. The *Internal Architecture Guide* explains the innards of the p-System and includes descriptions of the p-machine, the internal file structure, low-level I/O, and operating system functions. Finally, the *Assembler Reference Manual* describes 8086 assembly language.

Hobson's Choice? Here are some recommendations about which Pascal to choose.

If your application demands that your program run under PC-DOS, then IBM Pascal is your only option. It's a system-oriented language with many extensions and enhancements, but you may find its performance somewhat unpredictable. It is not a good choice for novice users. Our recommendation: use it if you have to.

UCSD Pascal is a less extended but more predictable language. Its accompanying operating system, the p-System, is easy to use and specifically oriented toward Pascal program development. The documentation is both clear and comprehensive. Our recommendation: use it if you can. ▲

```
XXXX: 0260 00
-EDS: 25D
XXXX: 025D 22.32 02.01 2B.37
XXXX: 0260 00.05
-WDS: 100 1 AF 1
-Q
```

If at any time the data from the system does not match that printed above, type Q to return to DOS and start over. Once you have a patched disk, try running the following program to see whether you have applied the patch correctly:

```
program test (input,output);
  var
    i,t:integer;
    b:array[1..10] of record
                        b1:integer;
                        b2:integer
                      end;
  a,q:array[1..10] of integer;
begin
  for i:=1 to 10 do
    begin
      b[i].b1:=i;
      b[i]b2:=i mod 2;
    end:
  if b[9].b2=1 then
    writeln('Ok.. Patch Applied Correctly')
  else
   writeln('Patch NOT Applied Correctly');
  t:=7; i:=1;
  a[i]:=t; q[i]:=t*(t+4);
  if q[i]=77 then
   writeln ('Ok.. Patch Applied Correctly')
  else
   writeln('Patch NOT Applied Correctly')
end.
```

▲

PUT YOUR IBM PC TO WORK WITH

THE DATA REPORTER

With the Data Reporter you receive an advanced user-friendly data management tool that delivers more power and flexibility than any other database system available for the IBM Personal Computer. This information management package consists of a unique combination of high speed machine language routines and BASIC programs. The modified "Program Generator" set-up program easily customizes the package to handle your data your way. Powerful search and sort routines let you manipulate your files to your requirements. The revolutionary "Soundex" search routine is included so you do not have to know the exact spelling of the data you are trying to access. Special features include fast data entry and update, calculations, data analysis, label maker, merge, data and program file back-up, reformat, global editing, and more. The report generator has many word processing features which allow you to print reports, form letters, and data in a variety of ways, with or without calculations. The Data Reporter uses single keystroke, menu driven commands with easy to follow prompting; non-programmers will have no problems using this powerful package. Sophisticated error avoidance and recovery techniques prevent undesired results. The BASIC language programs can easily be modified for those desiring special applications, while the machine language routines provide unexcelled speed. Use The Data Reporter for any of your data handling needs.

The program innovators at Synergistic Software will bring you this and other helpful, as well as entertaining, IBM PC software.

Send for detailed information to:

Synergistic Software

830 N. Riverside Dr., Suite 201
Renton, WA 98055
206-226-3216

newspeak

Δ Inherit the Wind. Blooming out of the arid western flatland of Medicine Bow, Wyoming, is a wind turbine that looks like one of those oversized sculptures by Claus Oldenberg. This one would be titled "Propeller on a Stick." While it looks a little funny from a distance, like a big pinwheel, up close it's awesome—262 feet high, big as Godzilla. A helicopter looks like an insect in comparison.

The WTS-4 (wind turbine system) is a project of the Hamilton Standard division of United Technologies. Its fiberglass blades are scheduled to start spinning at thirty revolutions per minute this September. The U.S. Department of the Interior Bureau of Reclamation is behind the building of the four-megawatt horizontal-access wind turbine. The first of its size in the world, it's the culmination of Hamilton Standard's wind turbine activity, which has been underway since the early 1970s.

The wind machine, with its expanded name, WTS-4 SVU (system verification unit), is part of an evaluation of technical and economic feasibility of wind turbines. Preliminary studies have indicated that wind turbines may be cost effective when coupled with existing hydroelectric installations.

The heart of the WTS-4 SVU is a set of Intel Model 230 microcomputers. One is located in the base of the steel tower and one is in the nacelle, the house-sized box behind the blades. Completely automatic, the micros control the turbine's operation, from making prestartup checks to diagnosing electrical faults and lubricating the system. Pitch control of the blades, to optimize energy capture, is one of their prime duties. Sensors in the nacelle send information on rotor speed and shaft torque to the 230s, which then adjust the angle and direction (into the wind) of the blades.

Too much wind may spin the blades too fast, causing the generator to go out of synchronization. The 230s will feather the blades to a slow stop when this occurs, starting them back up again when conditions have stabilized.

The WTS-4 SVU turbine in Medicine Bow is an upgrade of a three-megawatt unit being installed in the town of Maglarp, Sweden. Both are built by Hamilton Standard.

"Wind energy could represent up to 10 percent of new electrical energy resources in the next fifteen to twenty years," says Robert Gregoire, business development and marketing manager for Hamilton Standard. "Among the current alternative energy sources being considered—solar, geothermal, and photovoltaic—the wind is preeminent."

The Department of the Interior estimates that a wind farm of forty turbines could generate four hundred million kilowatt hours of electricity a year. Gregoire does some quick calculating. "That's enough electricity to run approximately fifty thousand homes."

Δ The Dope Syndrome. An interesting phenomenon occurs when you slip a pure silicon crystal into a beam of neutrons from a nuclear reactor. Most of the neutrons pass through the crystal as if it weren't there. But a few are captured by atoms of silicon 30 (one of three silicon isotopes commonly found in a sample), causing the silicon 30 to decay into phosphorus atoms.

The process is called neutron transmutation doping (NTD), and after the sample is removed from the neutron beam (and the radioactivity dies away) the result is a piece of silicon uniformly doped with phosphorus. The more phosphorus you add, the less electrical resistance there is and the greater the conductivity of the silicon.

A resistivity of zero, where the flow of electrons in a material is completely unhampered, has been achieved in certain materials (dubbed superconductors) at very low temperatures. The problem of excess heat given off by the movement of electrons is virtually eliminated, and the size of computer circuits made with this material can be reduced dramatically. A computer equivalent in power to an IBM mainframe made with superconductors would be about the size of a Rubik's Cube. But these theoretical super supercomputers are a few years down the road.

NTD will never completely replace today's industry standard, boron doping of silicon, for a number of reasons.

"It'll be hard to convince the computer industry that uniform phosphorus doping would be a benefit," explains John W. Farmer, senior research scientist at the University of Missouri in Columbia, Missouri. "The industry is committed to a certain technology and sees no need to change."

Another drawback is that NTD doesn't bring resistivity down to the levels needed in modern computer circuitry. Leaving the silicon in the flow of neutrons from the reactor long enough to reach those low resistance levels is not economically practical.

Bad times in the nuclear research industry constitute another problem that won't go away easily. No new research reactors have been built since 1963, and conventional nuclear power plants are unsuited for the process because they are designed to produce steam rather than to provide access to neutrons. Research reactors and commercial power reactors alike have a finite lifetime of perhaps fifty years.

Neutron doped silicon is perfect for electronic devices that handle high voltages or high currents in high-power machinery, like electric locomotives. One of the first applications of the technology went into electrical transmission lines in Europe. NTD at the moment seems more suitable for power control devices than for computers.

Nonetheless, Farmer believes that NTD may help in the development of high-speed, large-scale integrated circuits. The density of components is so high that a very uniformly doped material is needed, and NTD is more reliable than traditional methods.

"From that point of view, neutron transmutation doped silicon is much better as a starting material," Farmer explains.

At the moment, worldwide production of neutron transmutation doped silicon is around fifty metric tons a year. It is used mainly in Europe and Japan. The University of Missouri in Columbia is the only commercial source of the material in the United States. They produce about twelve metric tons annually.

Δ Here's a Knocking Indeed! For more than three hundred years, calculus has held a preeminent position in the mathematics curriculum. Deemed one of the finest achievements of the human mind, calculus has been the chamberlain of science since the days of Francis Bacon.

Discrete mathematics is knocking on the gate. Up to this time, freshman and sopho-

more mathematics students in college have been required to take a year of calculus. Recently a growing number of mathematicians and computer scientists have started to push for a more balanced curriculum featuring courses in discrete mathematics, including logic, probability, and the methods by which things combine and are counted.

Calculus is the tool of continuous mathematics, which posits nature as a smooth, unbroken flow, exemplified by the movement of objects in space. Discrete mathematics attempts to define nature in terms of individual elements, like the leaves on a tree. Computers are discrete machines, dealing with individual countable things.

Stephen White, director of special projects for the Alfred P. Sloan Foundation in New York City, defines calculus's traditional role as a gatekeeper. Discrete mathematics may soon share that role.

"There's been considerable discussion about it," White says. "People going into computer science are much more interested in discrete mathematics."

Last month the Sloan Foundation sponsored an invitation-only conference for mathematics educators at Williams College in Massachusetts. "It seemed that a general agreement was reached at the conference that changes would be made in the undergraduate curriculum," White reports.

Gale Young, a mathematics instructor at the University of Wyoming in Laramie, attended the conference at Williams College. "I think at the very least there'll be an alternative track for teaching freshman mathematics," he says.

"Now that computer science majors account for a larger part of enrollment, we're able to listen to them. But it's not just those students that need early exposure to discrete mathematics. Students in engineering and the physical sciences who expect to use computers should have experience with discrete mathematics very early."

How soon might we expect this shift in emphasis to become widespread? Ten years? "It'll be quicker than that. Has to be," explains Young. "It'll be more like five years, which is very quick for academia."

One problem holding up the teaching of discrete mathematics at the freshman level is a lack of proper textbooks. Help is on the way from John G. Kemeny, who coauthored in 1955 with J. Laurie Snell and Gerald L. Thompson the first introductory-level textbook for discrete mathematics, *Introduction to Finite Mathematics.* The Sloan Foundation has given Kemeny, a professor of mathematics at Dartmouth College, a grant to rewrite the book emphasizing the link to computers.

"There is now substantial support for a recommendation that a group of us made twenty-five years ago that the introductory mathematics sequence should be partly calculus and partly finite (discrete) mathematics," Kemeny says in a recent Los Angeles *Times* article.

Another leading supporter of giving discrete mathematics more emphasis is Antony Ralston, a mathematician and professor of computer science at the State University of New York at Buffalo. In the same article, he sums up the situation: "The rise of computers is changing the face of the world not just in obvious ways, but in some very significant intellectual ways.

"The overwhelming number of new problems which will come to mathematics to be solved come from areas which are related to computers and computer science."

Δ **The Incredible Shrinking Computer.** Don't be surprised if someday soon you have to wait outside a telephone booth while the person inside communicates by modem utilizing a portable computer. Panasonic, Sinclair, Quasar, and now Olympia USA are marketing miniscule but versatile computer systems that can be easily carried and used just about anywhere.

The briefcase-sized Olympia Portable Computer (OPC) comes complete with a 6502 microprocessor, a calculator-sized keyboard with sixty-five characters, a liquid crystal display panel, and an internal battery pack. Standard models come with 4K programmable memory, but the portable micro supports up to 52K of RAM and 64K of ROM.

An acoustic modem and RS-232C interface are optional, though they seem crucial to the concept of a portable computer. More standard features are a clock/controller for keeping track of appointments and a file system for storing information such as phone numbers.

Some of the programs available for the OPC are *Porta Budget, Porta Writer,* and *Porta Calc.* Snap (a derivative of Forth) and Microsoft Basic are available for programming applications. The software capsules are ROM units that plug into the back of the OPC. Because of internal memory restrictions, it's only possible to access one capsule at a time. The software is authored by an outside company, Friends Ami.

The OPC should be available in retail outlets this month; the carrying case is standard with every unit.

How well portable computing will catch on is not yet clear. The OPC has been received favorably at several trade shows, including Comdex in Atlantic City. But Panasonic and Quasar, which have had comparable products on the market for a few months, are apparently having trouble selling units. The portable microcomputer may take a while to reach its potential in the marketplace. It's a revolutionary concept worth keeping an eye on. ▲

What they're saying about **DISKLOOK,** the complete diskette displayer:

"Wow! That ***DISKLOOK*** is a real addition to DOS!" — Mike Todd, president, DC User's Group

"I wish I saw more programs this good; you should charge more for it!"— Jack Crone, editor, **PC Age**

"Thanks for writing ***DISKLOOK;*** we get good use from it at PC!" — Jim Edlin, founding editor, **PC** mag.

Two programs no IBM/PC should be without:

DISKLOOK — a full-featured diskette display program; shows hidden files, erased files, full sector displays, maps storage use and file location, and more. **$20**

UNERASE — recovers erased diskette files, even when partially over-written. Can you afford to be without ***UNERASE***? **$20**

More utilities for the advanced IBM/PC user:

FILEHIDE — hides and unhides files, protects files from erasure **$10**

SECMOD — reads and easily modifies diskette sectors, in hex and ASCII; a patcher's dream **$20**

THE NORTON UTILITIES

For all orders, add $5 for diskette postage & handling; California 6% tax

Peter Norton 1716 Main Street, #E, Venice, California 90291

top shelf

Software for the IBM PC and the Apple II

With spoken instructions, our tested software doesn't try your patience.

EASY Every Alpha software package contains spoken instructions on cassette to guide you through the program on the disk. You get professional software that a novice can use.

PROVEN Our business software for the IBM PC has been used since 1979 on the Apple. Thousands of people agree that our versatile **Data Base Manager** and **Mailing List** programs are effective time- and money-saving devices.

INNOVATIVE **Question** is a fun game based on artificial intelligence. It tries to guess what city, person or animal you're thinking of, and gets smarter as you play. You can even teach it new topics.

Type Faces lets your dot matrix printer generate 15 different kinds of large lettering.

With **The Apple-IBM Connection,** you'll be able to transfer files from the Apple II to the IBM and vice versa. Your VisiCalc work can be transferred without retyping or errors. Requires no technical knowledge. The software does all the work for you.

Alpha software products available at participating **ComputerLands.**
Or call us for your nearest dealer: 617 229-2924

alpha SOFTWARE CORP

6 New England Executive Park • Burlington, Massachusetts 01803

Data Base Manager: $185 Mailing List: $95 Question: $45 Type Faces: $95 The Apple-IBM Connection: $120

SYSTEM NOTEBOOK

by Alan Boyd

Before we continue on our journey through the software side of the pc, we have to clear up some issues raised by recent developments. When the first article in this series appeared (June 1982), anyone who purchased an IBM Personal Computer and selected DOS as his operating system was presented with Version 1.0 of DOS. Now, however, IBM has introduced Version 1.1.

For the time being, we'll assume that not everyone has upgraded their systems to Version 1.1, although we highly recommend that you do. IBM is making it as painless as possible to upgrade by maintaining the low ($40) price for the new and better product.

As a product such as MS-DOS (the generic name for IBM's PC-DOS) matures, new enhancements and features will be added and minor problems or irregularities will be fixed. The most notable improvement in Version 1.1 is its ability to handle the new double-sided, double-density disk drives. Other improvements include an enhanced copy facility, several enhancements to the mode command, and file time-stamping. There are also several speed improvements.

Because some users may not want to change the status quo by switching to Version 1.1, we will make every effort to address this series to users of both versions. Where there are differences, we'll examine both systems.

On that note, let's move right back into the stream where we left off last month.

By now we've seen how to perform all the basic functions associated with DOS. These include formatting blank disks, making backup copies of entire disks, copying files from one disk to another, changing the names by which files are stored, deleting files, displaying the contents of a file, and reading the directory of a disk.

These are all the functions that everyone who uses the pc will have to learn at some point. In computer jargon these commands do what are called housekeeping tasks.

The next part of our series will focus on areas that are not required knowledge for pc users but that nevertheless are of use to anyone who takes the time to learn them.

Timing Is Everything. The first of these areas concerns the pc's ability to handle date and time-related functions. Timing is a vital part of any type of computer equipment, and the pc is no exception. Today's microcomputer chips and their related circuitry operate at speeds that are quite literally running up against the physical boundaries of space and time. The numbers cited for the timings of these chips are often staggering and will probably mean little to the average user.

In many computers and operating systems, timing functions are carried out without the user knowing about them. In the pc, however, some of the characteristics of the internal timing have been left open to user manipulation. This means that the user can access timing information via software and can use this information in programs or have programs take control of the pc at specific times.

DOS, which after all is simply a program running on the pc, makes use of the timing information in some interesting ways. In the Command Interface, for example, this boils down to two external DOS commands—*date* and *time*. Before we look at the commands in depth, though, let's briefly examine how the pc relates to time in general.

The heartbeat of the pc is produced by an internal clock—a crystal timer, to be exact. This clock produces a pulse with a frequency of 14.31818 megahertz. That is, it pulses more than fourteen million times per second. This is the heartbeat from which all internal timing information is derived.

Internal circuitry divides this pulse so that it can be used to drive various devices at different, although related, speeds. One subdivision of the 14.3 MHz crystal pulse is used to drive the Intel 8088 microprocessor chip that is the brain of the pc. The crystal pulse, or clock signal, is divided by three to produce a 4.77 MHz signal used to drive the microprocessor. The clock is also used to generate the 3.58 MHz pulse (14.31818 MHz divided by four) that produces the color burst signal required to drive a color television, RGB monitor, or composite color monitor.

The main clock is supported by a chip known as a three channel timer/counter (an Intel 8253-5, to be precise). Channel 1 of this counter is used to maintain an accurate time of day. In effect, this capability is tantamount to having a very accurate real-time clock built into the pc.

The minimum accuracy of this clock is 1.05 microseconds. This use of a super high-speed counter simply to keep track of the time of day may seem like overkill, but it does give the pc the ability to perform tasks based on time, and it allows DOS to keep track of when files are written to disk.

Let's take a more detailed look at the *time* and *date* commands.

Time. *Time* is an external DOS command that's used to display and set the internal system time information. The ways in which DOS uses time information have changed dramatically in Version 1.1. In Version 1.0, the internal time is set to zero when the system is booted. The internal timer therefore acts as an elapsed time indicator, tracking the amount of time that's passed since the system was booted. DOS Version 1.0 itself never uses the internal time, although any application program running under DOS can make use of it.

DOS Version 1.1, however, makes extensive use of this information. The rest of this discussion, therefore, only applies to Version 1.1 and any later versions that may be released.

DOS uses the timing information to stamp files. Whenever you create a file and write it to disk, DOS enters the system time of file creation in the directory along with the name and other information about the file. When the file is subsequently modified, the time stamp is also modified.

This time stamping is particularly helpful for those who have large storage devices attached to their computers. Although IBM hasn't yet announced a large mass storage device for the pc, it is generally anticipated that they will eventually market a Winchester-type hard disk; after all, they pioneered Winchester technology. Several other manufacturers are already offering such devices, and many people are already using them.

One of the problems of working with a large storage device is that the medium is not

usually removable. Therefore, it is extremely difficult to make backup copies of the information. Several solutions exist for this problem, ranging from the use of a second hard disk (expensive and not very cost-effective) to backing up onto floppy disks (cost-effective, but very time consuming).

The time stamp facility makes it quite easy for a programmer to create an incremental backup system, one that can be used to back up only those files that have been modified since the last backup. Such a system would compare the date and time stamps of the file that is to be backed up with the date and time stamps of the last backup of the file. If they are the same, then there's no need to make the copy. If they differ, the copy program would proceed.

The disk-resident *time* command is used both to display and to alter the internal system time. It is invoked simply by typing the command at the keyboard with the DOS disk in the drive. DOS will read the file Time.com and display on screen the current system time. It will also prompt you to enter a new time if you wish. Here's an example:

```
> TIME
Current time is 07:10:55.51
Enter new time:
```

The time displayed is in the format, *hours:minutes:seconds.fraction of a second*, with the hours on a twenty-four-hour scale. Obviously, with a resolution of microseconds, it's quite easy for the pc to maintain the time down to the hundredth of a second.

If you simply press the enter key in response to the new time prompt, the internal time will not be changed. If you wish to set the time to something else, then enter it as

hh:mm:ss.xx

The internal time will be updated to whatever you enter as soon as you press the enter key. There is no need to enter the full time down to the fraction of a second. If the time is 3:15 in the afternoon, all you need to enter is 15:15. If it's exactly four o'clock, then simply enter 16. DOS will accept any part of the time and leave the precision up to you.

Date. *Date* is an external DOS command that is used both to display and to change the internal system date information. *Date* is used to stamp files created under both Version 1.0 and Version 1.1 of DOS.

When you first boot the system, DOS prompts you to enter the date. Since all files created under DOS are date stamped, it's advisable to enter the correct date. Failure to do so may lead to unpredictable consequences with some application software packages.

DOS only understands dates that are later than January 1, 1980. Of course, there were no DOS systems running prior to that date. If for any reason you wish to display or change the internal system date, enter the *date* command and DOS will display the current date and give you the same prompt it always gives when you boot the system. If you wish to modify the date, enter the new date in the format *mm-dd-yy* or *mm/dd/yy*.

The upper limit for dates is 12-31-99, which limits the useful life of DOS to the next seventeen years. Fortunately, there will be a better version of DOS available long before the year 2000.

Unlike the *time* command, *date* requires a complete entry; the system will not accept a partial date.

Sys. That takes care of the *time* and *date* commands. The next subject we'll examine is the *sys* command.

The DOS manual describes *sys* as a command to transfer the operating system files from one drive to another. This description has led to quite a lot of confusion. As you know, the *format d:/s* command both formats the target disk and writes the system files onto it. So why is there a need for the *sys* command?

Most people have assumed that the purpose of *sys* is to convert a nonsystem disk—one that was created by the *format* command without the */s* switch—into a system disk. However, when you attempt to use *sys* to make this conversion, you get the error message

No room for system on destination disk

What function then does the *sys* command perform? To understand its raison d'etre, we need to make a little foray into copyright law.

DOS is a copyrighted piece of software. It is copyrighted first by Microsoft Corporation, the owner, and then by IBM, the licensee. No other person or company is allowed to distribute DOS without first arranging a distribution license from IBM (who currently do not offer such a license).

As a consequence, it is not legal for a software manufacturer to distribute copies of the operating system on a program disk. Many software distributors, however, want to make it as easy and painless as possible for you to install a newly purchased software package on your machine. The best method to accomplish this is to distribute a disk that has been formatted with the *format d:/s* command but from which the system files have been erased.

By preparing the disk in this way, the software manufacturer remains in compliance with copyright law. But he also leaves room in tracks 1 and 2 so that you can transfer the DOS system files yourself. The *sys* command was provided expressly to allow you to make that transfer.

When you buy a new piece of software that runs under DOS, the procedure you'll

generally follow to make the disk bootable is as follows:

1. Format a blank disk with the simple *format* command.

2. Copy the purchased disk onto the newly created disk, using the *diskcopy* command.

3. Transfer the system files from your DOS master disk to the new disk, using the *sys* command.

4. Place the purchased disk in a safe place and use only the backup. Never write anything onto the purchased disk.

There are other ways to achieve the same result, but sometimes other methods may not work correctly. Before transferring system files, consult the documentation that comes with your software package. Certain software packages, including some that use sophisticated software protection schemes and don't allow you to make backup copies, do not require DOS. Transferring files via *sys* to such a protected disk may cause irreparable damage, so, for safety's sake, read the documentation first.

The Program of Dorian Gray. Another creative use for the *sys* command is to update the operating system on a disk from an older version to a newer one. If you had previously created several DOS master disks using Version 1.0 of DOS and you wish to upgrade them to the Version 1.1 that you just purchased, you may do that with the *sys* command. In this way you could alter all of your old master disks so you use only the most recent version of the operating system.

Diskcomp. The *diskcomp* command is a very useful DOS utility. An external command stored on the DOS master disk, *diskcomp* is used to compare the contents of two disks to see if they are the same.

Usually you'd invoke *diskcomp* after you had made a backup of a disk via *diskcopy*; *diskcomp* would verify that the copy was completely successful.

Diskcomp compares track by track rather than file by file. It reads whole tracks of data into memory from each disk and then compares them. If discrepancies are found on any track, the bell will ring and the offending track number will appear on the screen. *Diskcomp* can be used with either single-drive or multiple-drive systems. The syntax is:

DISKCOMP D: [d:]

where the second *d:* is optional. If only one drive designation (or no drive designations, in which case the command will default to the logged drive) is entered, then *diskcomp* will perform a single-drive comparison, prompting you to swap disks. If two drive designations are entered, the command will perform a dual-drive comparison.

Diskcomp is an excellent command by which to illustrate some of the concepts we've already discussed. If possible, try the following example on your own computer.

The disk that we'll use is the DOS Version 1.0 master disk, although a similar scenario could be created with virtually any disk. Place a DOS master disk in drive A and boot the system. Enter the proper date, and DOS will display the A>prompt. Next, format a blank disk in drive B, using the *format b:* command. When the formatting is complete, copy the entire contents of the master disk, which should still be in drive A, to the new disk in drive B, using the command *diskcopy a: b:*. When you're done, type:

A>DISKCOMP A: B:

The computer will respond with

Insert first disk in drive A
Insert second disk in drive B
Strike any key when ready

Press a key, and the compare procedure will start. If the *diskcopy* command has performed correctly, you should get the response

Diskettes compare ok

If you don't get this message, repeat the process until you do. If it consistently fails, there is probably something wrong with your disk drives and you should consult your dealer.

Now let's deliberately change the contents of the duplicate disk. There should be a file on the disk by the name of Comm.bas. Look at the file contents with the *dir* command to be sure. Then delete this file from the duplicate disk, by typing *erase b:comm.bas*. When you've done this, invoke the *diskcomp a: b:* command again.

This time, it will return the message

Compare error(s) on track 00

This message indicates a discrepancy on track 0. However, since the Comm.bas file was the very last entry in the directory, it is highly unlikely it is stored entirely on track 0—particularly in light of the fact that track 0 is used to store the directory and operating system files.

Let's go a little further. Delete all the files from the duplicate disk by typing *erase b:*.*. Then perform the comparison again by typing *diskcomp a: b:*. Again, only one compare error will result.

The FAT's in the Fire. Here we have proof of the fact that the DOS *erase* command never actually erases data from a disk; it simply marks it in the directory as erased and updates the File Allocation Table to indicate that the sectors occupied by the data are available for use by other files. Since the directory and File Allocation Table are maintained on track 0, this is the only area where the *diskcomp* operation will find differences between the two disks; hence, the result we saw.

The *diskcomp* command is an excellent tool for verifying the integrity of a backup operation and should be used directly after the backup. If files are deleted from a disk, however, *diskcomp* will only show that the directory and File Allocation Table have been changed.

Comp. DOS provides *diskcomp* as a verification procedure for backups created with the disk-oriented *diskcopy* command. It should be obvious then that the *comp* command is the complement of the file-oriented *copy* command.

Comp is an external DOS command used to compare individual files. In general, *comp* is used immediately after a backup copy of a file is generated, as a verification of data integrity. The files compared may be on the same or different disks. The syntactic format for *comp* is:

COMP D:filename d:filename

The first drive designation (D:) is mandatory if the file isn't on the currently logged drive. The second drive designation is necessary if the second file doesn't reside on the currently logged drive. If the second file name isn't entered, DOS will assume that you want to compare two files with the same name. In this case, of course, the two files must be located on separate disks, since two

Let Your IBM PC Train You To Read Faster & Understand More

THE SPEED READER

Contains 2 disks with 5 separate programs and 34 reading selections designed and tested by reading specialists to improve speed and comprehension.

THE SPEED READER is now available at your IBM dealer or send check or money order for $74.95 to Davidson & Associates, 6069 Groveoak Pl. #14, Rancho Palos Verdes, CA 90274. (California residents add 6½% sales tax.) We'll send you your software by return mail, and we'll pay the postage. SATISFACTION IS GUARANTEED! If you are not completely satisfied with this product, send it back to us within 7 days for a full refund minus $10 handling charge.

files with the same name cannot exist on a single disk.

The wild card characters * and ? can be used with *comp*, but a multiple file compare will not be done. Only the first set of files that match the ambiguous file names will be compared.

For those who are intimidated by having to enter long command lines, the *comp* command has an easy mode. By simply entering the *comp* command, a user will be stepped through the process of entering the necessary information. DOS will respond with

Enter primary file name

Type in the name of the original file and press the enter key. The computer will then prompt

Enter 2nd file name or drive id

Type in the name of the backup copy, along with the drive designation letter and colon if the backup is not located on the logged drive. If the file has the same name as the original file, then there's no need to retype the name of the file. Simply enter the drive designation and colon and press enter.

Comp will then read the first file into memory, or part of the file if it's too big to fit, and then a similar part of the second file. It will then perform a byte-by-byte comparison of the two files. If the files are obviously different, the *comp* process will be halted. The first indication of extreme differences between the two files is, of course, different file lengths. *Comp* first checks to make sure the two files are the same in length. If they aren't, DOS returns the error message

Files are different sizes

and then prompts the user to enter other file names. Hitting enter at this point will get you out of the *comp* command.

If *comp* finds differences in the two files, it displays them on the screen. The display format for such findings is

Compare error at offset 9E
File 1 = 44
File 2 = 2E

The offset is the number of bytes after the beginning of the file where the discrepancy was discovered, and the two numbers after the File 1 and File 2 designations are the contents of the two differing bytes. All the numbers are in hexadecimal notation—that is, base sixteen.

Should *comp* discover ten discrepancies between the two files, it concludes that the files are radically different and aborts the compare process. In this case, the message

10 Mismatches - aborting compare

is displayed.

It just happens that there are two files on the DOS master disk that have the same length but different contents. Both files contain music information for the sample programs contained on the disk; their names are Pop.bas and Forty.bas. To compare them, enter

COMP POP.BAS FORTY.BAS

and note the results.

Any time an important file or an entire disk is backed up, such as at the end of the day in an accounting department or after a master *VisiCalc* template has been prepared, it is worth the few seconds it takes to verify the integrity of the backup, using either *diskcomp* or *comp*. If for any reason the compare operation reveals errors, you can recopy the disk or file and retry the comparison. The safety afforded by this procedure is worth the few moments it takes to do it, particularly if the file contains vital information.

For those who have previously used CP/M, these commands—*comp* and *diskcomp*—are analogous to the [V] (verify) option that can be tagged onto CP/M's *pip* command. They perform substantially the same operation.

The way that DOS handles the process may seem the longer of the two methods since you can copy and verify a file under CP/M with a single command. However, DOS has its own facility to allow the user to set up streams of commands and store them in files on disk so that an unsophisticated user can easily submit long command strings without having to reenter them every time. This "batch" facility will be our topic next month. ▲

For Quality Software • Multiple Entry Modes • Comprehensive Documentation

OMRIC

SCREEN MACHINE™

Makes possible high-speed cursor-positioning, scrolling, erasing, and text handling for users of Pascal, FORTRAN, COBOL, and Assembler. (For IBM-PC)

SPACE GUARDIAN™

Traversing the local galaxy to search out and destroy alien spacecraft, you will need logistic and tactical skills in your crucial mission to save planet Earth. (For IBM-PC)

GENTERM™

Conforming to most protocols involving character-by-character or line-by-line communication, this package can equip your computer system to communicate with other micro-, mini-, or mainframe computers. (For IBM-PC, Osborne I, or any CP/M Computer)

BLINGSPLATZ™

Real-time arcade game named for the eerie horde of intergalactic creatures whose relentless attack on your space patrol will challenge all your skills. (For IBM-PC)

CHAMPION DRAUGHTS™

How about playing checkers against the wily computer? Don't worry... the caliber of the computer's play can be selected to match your own. (For IBM-PC)

OMRIC Corp., 1268 Main St., Newington, CT 06111

Please identify the nearest distributor of:
☐ Screen Machine™ ☐ Blingsplatz™
☐ Space Guardian™ ☐ Champion Draughts
☐ Genterm™

Name_____
Address_____
City_____ State_____ Zip_____

1268 Main Street, Newington, Connecticut 06111
Tel (203) 666-4240

marketalk news

Unless otherwise indicated, software listed runs in DOS on machines with either display adapter and requires 64K and at least one disk drive.

Δ New from **IBM** (Box 1328, Boca Raton, FL 33432; 305-998-2000): A 64–256K memory expansion board. $475 in minimum configuration; additional 64K modules $200 each. Δ A prototype card for engineers and hobbyists. $45. Δ *Decathlon*, a program by Microsoft that simulates a ten-event track-and-field competition. Up to six people can participate. $35.

Δ **Squire, Buresh Associates** (4 East Central Street, Suite 2029, Worchester, MA 01613; 617-832-6430) offers *Repair*, a utility that allows you to put a file on disk by track or sector, hide files or find hidden files, resurrect erased files, and change a directory. $59.95. Δ *Data Design* is a relational database with full support for hard disk drives. It can be customized with an optional Query language. A product of **Insoft** (10175 S.W. Barbur Boulevard, Suite 202B, Portland, OR 97219; 503-244-4181). Printer recommended. $225; $300 with Query.

Δ **Corvus Systems** (2029 O'Toole Avenue, San Jose, CA 95131; 408-946-7700) sells the Omninet interface, which allows networking of up to sixty-three machines. Price varies.

Δ **Sapana Micro Software** (Box 748, Quincy, IL 62301; 217-224-4337) publishes the *Software/Hardware Directory and Guide for IBM Personal Computer*. The directory is semiannual with monthly updates. Six-month subscription: $10. Sample copy free.

Δ Four programs from **Data Consulting Group** (877 Bounty Drive, Foster City, CA 94404; 415-349-4001): *Mail List/Label*, a mailing program with a special feature—you can search and display for the exact or the most similar record. Requires 48K and two drives. $95. Δ *Screen Generator*, which allows you to save a screen as keyed. Requires 48K. $45. Δ *Programmer Utilities*, a menu-driven package that allows you to display, edit, or list virtually any file. Requires 48K. $75. Δ *Retail Sales/Inventory*, which includes an invoice-formatted screen and inventory and customer credit checks. Requires 48K and two drives. $195. Δ *Telephone Directory*, which keeps birthdays and personal notes, as well as name, address, and two phone numbers. Requires 48K and two drives. $25.

Δ **Design Enterprises of San Francisco** (Box 14695, San Francisco, CA 94114; 415-282-8813) publishes *Write, Edit & Print*, a book that shows how to select and use a computer for word processing. Included is an annotated program listing. Paperback $24.95; clothbound $34.95. Program disk $29.95 extra.

Δ **Giltronix** (970 San Antonio Road, Palo Alto, CA 94303; 415-493-1300) offers a line of switches that will allow, for example, several computers to share one printer or one computer to use the same port for both a printer and a modem. $149–$899.

Δ *Quality Management Monthly* is a newsletter "dedicated to the assurance of high quality throughout computer software system development." Published by **Software Research Associates** (Box 2432, San Francisco, CA 94126; 415-957-1441). Subscription: $225 in the United States and Canada; $275 elsewhere.

Δ **Davong Systems** (1061 Terra Bella Avenue, Mountain View, CA 94043; 415-965-7130) offers a 64K expansion board (expandable to 256K) with two serial ports. $495.

Δ New from **Ensign Software** (2312 North Cole Road, Suite E, Boise, ID 83704; 208-378-8086): *String Sort*, a utility designed for use with Basic that will sort any string array into alphabetical order. Fits in 512 bytes of memory. $24.95. Δ *Electronic Disk*, which allows RAM to be used as a disk drive or printer spooler. Requires 128K. $69.95. Δ *Indexed Sequential Access Method*, a filing program that sorts 5,000 records in twelve seconds. Requires 48K. $69.95. Δ *Cross Reference*, which indexes a program listing by variable and line number. Requires 48K. $24.95. Δ *Games Package 1*, including ten strategy and educational games. Requires 48K. $29.95. Δ *HP7470 Plotter Demo*, with nine demonstration programs for the HP7470 or HP7225 plotter. Requires 48K. $24.95. Δ *Bowling League Statistics Manager*, capable of keeping statistics for several leagues simultaneously. $99.95.

Δ *Slynx* is a fast-moving arcade game from **Ivy Research** (88 Yale Station, New Haven, CT 06520; 203-789-0676). Requires eighty-column screen. $34.95.

Δ Baby Blue is the name of a coprocessor card that allows the Personal Computer to use CP/M software. Includes 64K memory. Also available: *WordStar* and *MailMerge* preformatted for use with Baby Blue. A product of **Xedex** (1345 Avenue of the Americas, New York, NY 10105; 212-489-0444). $600. With *WordStar* and *MailMerge*, $980.

Δ **Software Dimensions** (6371 Auburn Boulevard, Citrus Heights, CA 95610; 916-722-8000) has adapted its *Accounting Plus* program for use with Baby Blue. 5M hard disk required. Price varies.

Δ **Professional Micro Systems** (Box 1377, Gardnerville, NV 89410; 702-782-8105) has announced a network system for use with its hard disks. The network can connect up to twenty-eight computers. A variety of CP/M and CP/M-86 software is available. Price varies.

Δ *IBM Personal Computer User's Guide* is a paperback combining reference material with a Basic tutorial. Available from **Osborne/McGraw-Hill** (630 Bancroft Way, Berkeley, CA 94710; 415-548-2805). $16.95.

Δ *Full Screen Editor* is intended for use with either programs or text files. It makes full use of the IBM keyboard. Produced by **Frank L. Schiele Software** (1375 Tobias Drive, Chula Vista, CA 92011; 714-420-9860). $55.

Δ **Corona Data Systems** (31324 Via Colinas, Westlake Village, CA 91361; 213-706-1505) offers a hard disk system with selective floppy backup—information can be chosen for backup on the basis of when it was recorded, by file or file category, or by other criteria. The hard disk can be substituted for a floppy drive in the system unit. 5M $1,995; 10M $2,495.

Δ **Electrohome** (809 Wellington Street North, Kitchener, Ontario, Canada N2G 4J6; 519-744-7111) has a new color monitor available in both medium and high resolution. Price to be announced.

Δ *Concurrent CP/M-86* is a single-user, multitasking operating system from **Digital Research** (Box 579, Pacific Grove, CA 93950; 408-649-3896). It allows you, for example, to use a printer, consult a database, and receive electronic mail, all at the same time. 256K recommended. $350.

Δ *Flash Disk* is a program that lets you use RAM to simulate an extra disk. Offered by **Seattle Computer** (1114 Industry Drive, Seattle, WA 98188; 800-426-8936). Free with purchase of a memory expansion board ($475 and up).

Δ *Ken Uston's Professional Blackjack* is a program that shares the secrets of the world's most famous blackjack player. Eighty-column screen required. A product of **Intelligent Statements** (Box 600, Holmes, PA 19043; 800-345-8112, 800-662-2444 in Pennsylvania). $89.95.

Δ *Basic'* (pronounced Basic prime) is a programming environment for Basic programmers. The *Basic'* editor provides automatic indentation and other features to eliminate the need for line numbering and enhance the readability of source code. The source is then translated to a standard Basic listing for debugging via the Basic interpreter. Cross-referencing to source code is provided during the translation. *Basic'* is available from **Delta Micro Systems** (Box 15952, New Orleans, LA 70175; 800-535-1814). $129.

Δ **Star Computer Systems** (18051 Crenshaw Boulevard, Torrance, CA 90504; 213-538-2511) has released three new business packages designed to run under CP/M or CP/M-86: An accounting package with general ledger, accounts receivable, accounts payable, and payroll. $1,250; individual programs $400 each. Δ *Property Management*, to keep track of rents and deposits, prepare financial statements, and print checks. $950. Δ *Legal Timekeeping and Billing*, which keeps track of attorneys' hours, costs, and billing. $950.

Δ Freedom Network is an electronic message service offered by **Graphnet** (329 Alfred Avenue, Teaneck, NJ 07666; 201-837-5100). The monthly fee is $5. Messages cost about thirty cents per hundred words.

Δ **Digital Marketing** (2670 Cherry Lane, Walnut Creek, CA 94596; 415-938-2880) has announced *Field Companion*, a program for traveling businessmen that keeps customer lists and sales information. Invoices may be output to printer or modem. Requires CP/M and 56K; CP/M-86 version requires 128K. $295.

Δ **American Training International** (3800 Highland Avenue, Suite 300, Manhattan Beach, CA 90266; 213-546-4725) offers a line of interactive training programs. Subjects include Basic, databases, *VisiCalc*, *SuperCalc*, *WordStar*, PC-DOS, and CP/M. 48K required. $75 each.

Δ **Columbia MicroSystems** (110 North Tenth Street, Columbia, MO 65201; 314-875-8900) sells a dual 8-inch disk drive system in a 2½-inch-high cabinet which fits neatly on top of your system unit. Capacity: 2.4M formatted. $1,750; $2,095 with driver and controller for PC-DOS and CP/M-86.

Δ *I-Plot* is a menu-driven plotting utility that can accept data from either keyboard or a data file. Requires 128K and a color adapter. From **Management Solutions** (17660 Millbrook Drive, Chagrin Falls, OH 44022; 216-543-9263). $250.

Δ *The Jeffries Report* is a newsletter that offers a monthly summary of events in the microcomputer field. Published by **The Code Works** (Box 550, Goleta, CA 93116; 805-683-1585). Subscription: $15 in United States and Canada; $20 elsewhere.

Δ *IBM/PC Survivor's Manual* is a forty-eight page introduction intended for complete novices. Published by **Mayer Associates** (755 New York Avenue, Huntington, NY 11743). $11.95.

Δ New from **International Computers** (13048 Olive Boulevard, Saint Louis, MO 63141; 314-878-8838): *Forth Computer Assisted Instruction Course*, as the name implies, is a computer instruction course for the language Forth. Requires 48K. $59.95. Δ *SpeedupPlus* is a disk speed-up utility compatible with PC-DOS 1.0 or 1.1, Basic, Fortran, both Pascal systems, and Forth. $199.95. Δ *Budget-Tax Deduction Record* keeps track of deductible expenses. $39.95.

Δ *GSS-Plot* is a plotting utility that can produce charts and graphs and output to most plotters, including the HP 7470A. Requires 40K and CP/M-86. (A PC-DOS version is scheduled for October.) Produced by **Graphic Software Systems** (Box 673, Wilsonville, OR 97070; 503-682-1606). $500.

Δ *Tiger Database* is a program suitable for keeping inventory, accounting, personnel information, and other applications. It produces reports, labels, and form letters, and includes a training disk. Requires two disk drives. Contact **Tiger Software** (3857 Birch Street, Suite 600, Newport Beach, CA 92660; 714-552-8650). $250.

Δ **Falcon Safety Products** (1065 Bristol Road, Mountainside, NJ 07092; 201-233-5000) produces Dust-Off II, a spray that removes dust from equipment. $24.95; refills $4.50. Δ Stat-Off II is an accessory that removes static charges from surfaces. $17.95.

Δ **Smiser & Associates** (Box 2095, Truckee, CA 95734; 916-583-9329) is a custom software firm specializing in construction and general accounting. Prices start at $750.

Δ **Vista Computer Company** (1317 East Edinger, Santa Ana, CA 92705; 714-953-0523) offers a 128–576K memory expansion card. $520 in minimum configuration.

Δ *Screen Print* is a screen dump program from **M.A.P. Systems** (1120 NASA Road One, Suite 444, Houston, TX 77058; 713-333-9640). It is intended for use with either an MX-100 printer or an MX-80 with Graftrax. $44.95. Δ The same company sells a color screen dump designed for the IDS Prism. $49.95.

Δ **Tarrtec Enterprises** (Box 81, Commack, Long Island, NY 11725; 516-543-6579) produces a security-coded name directory capable of storing 1,200 records per disk and producing mailing labels. Requires 16K. $100.

Δ *The Instructor* is a tutorial program designed to familiarize the user with Personal Computer hardware, operating system, and software. A product of **Individual Software** (24 Spinnaker Place, Redwood City, CA 94065; 415-591-4166). $60.

Δ **Sentient Software** (Box 4929, Aspen, CO 81612; 303-925-9293) has announced *Cyborg*, a text adventure game with sentence parsing. Written in Forth, it has its own operating system. Price to be announced.

Δ From **SuperSoft Associates** (Box 1628, Champaign, IL 61820; 217-359-2112): *Disk-Edit*, a screen-oriented disk editor for CP/M. Compatible with either hard disks or floppies, the program displays each byte in both hexadecimal and ASCII values. $100. Δ *Personal Data Base*, a menu-driven program designed for nonprogrammers, is suitable for keeping mailing lists, birthdays, and personal investment information. Requires 48K. $125.

Δ **DB/DC Software Associates** (945 Riverside Drive, Metheun, MA 01844; 617-682-6629) has a line of four programs: *Ultimenu*, a menu that allows the user to invoke up to thirty-six programs or batch files while still being able to use DOS commands. Requires 48K. $19.95. Δ A different version of the program that provides a Basic prompt instead. Requires 48K. $15. Δ *NewText1*, a text processor designed for long manuscripts that require a contents page and index. Also useful for form letters, mailing labels, and memos. $125. Δ *NewMacro*, a subset of *NewText1*, which allows macros to be kept in a text file. $65.

Δ Hardware from **Panasonic** (One Panasonic Way, Secaucus, NJ 07094; 201-348-7000): A hi-res monochrome display. $220. Δ Two parallel, dot-matrix printers, the Model KX-P1160 with a nine-by-thirteen matrix and the Model KX-P1090 with a nine-by-nine matrix. Prices to be announced.

Δ *Using VisiCalc* is a training program that includes two 120-minute audio cassettes and a data disk. Contact **McMullen & McMullen** (Box 230, Jefferson Valley, NY 10535; 914-245-2734). $59.95.

Δ **Engineering-Science** (57 Executive Park South, N.E., Suite 590, Atlanta, GA 30329; 404-325-0770) produces *Plotrax*, a plotting and statistics package. Requires color adapter and Epson MX-100 or MX-80 with Graftrax. $97. Δ *Transplot*, a subset of *Plotrax* with machine language screen dump. $27.

Δ From **Eagle Software Publishing** (110 West Lancaster Avenue,

Wayne, PA 19087; 215-964-8660): *Money Decisions*, a package of thirty-four business and financial programs. $199. Δ *Tax Decisions*, a program for tax professionals that allows the user to analyze up to five different strategies at the same time. $299.

Δ **Spectrum Software** (690 West Fremont Avenue, Sunnyvale, CA 94087; 408-738-4387) offers *Personal Finance Manager*, an accounting program for home and small business. Requires monochrome monitor. $75 Δ *General Ledger* and *Accounts Receivable*, a double-entry system for business. $150 each or both for $250. Δ *Mathematics Series*, a menu-driven program that does statistical and numerical analyses and plots and solves systems of linear equations. $50.

Δ A forum on business applications for microcomputers has been scheduled for November 1–3 in Palm Springs, California. Contact **Enterprise Information Systems** (Box 1154, Greenwich, CT 06836; 203-629-3510).

Δ **SoftWhere!** (2162 Deerfield, Thousand Oaks, CA 91362) sells *Suite 16*, a module that allows you to select both system functions and applications programs from a menu and still give standard PC-DOS commands. Monochrome display required. $75.

Δ **Hourglass Systems** (Box 312, Glen Ellyn, IL 60137; 312-690-1855) produces three programs, all of which require two disk drives: *Fast Figure*, a three-dimensional spreadsheet including a full range of financial functions and all methods of depreciation. $150. Δ *Market Time*, an investment timing tool to help you spot stock market trends. $75. Δ *High Yield*, an investment management tool that calculates yields and performance. $75.

Δ *PEP Business System* is a package from **Performance Engineered Programming** (3970 Syme Drive, Carlsbad, CA 92008; 714-434-6023). It includes inventory control, accounts receivable, accounts payable, payroll, and general ledger. Also includes warranty. Requires two 320K disk drives; also compatible with Winchester disk. $495.

Δ *Datamate* is a multipurpose electronic filing system. It includes a user-defined function that can be changed to fit the individual application. Two disk drives recommended. From **Compu-haus** (1840 N.W. 50th Terrace, Kansas City, MO 64151; 816-741-2669). $299.95.

Δ The latest from **Norell Data Systems** (3400 Wilshire Boulevard, Los Angeles, CA 90010; 213-257-2026): *Easyedit*, a text editing system with a split screen mode that allows you to edit two files at once. $149.95 Δ *Easytext*, a text formatting program that can also create a table of contents. Intended for use with *Easyedit*, *WordStar*, or *EasyWriter*. Requires printer. $79.95. Δ *Easycalc*, a spreadsheet program that shares commands with *Easyedit*. $99.95. Δ *Easyproof*, a speller with a 50,000 word dictionary. Requires printer and two disk drives. $79.95. Δ *Visualize*, a package that produces hard copy reports from *VisiCalc* templates. Requires printer. $100. Δ *Disk Magic*, which allows the user to access a disk sector by sector. Displays data in ASCII and hexadecimal formats. Requires 32K and printer. $49.95. Δ *Computer Chef*, a program that stores recipes and calculates ingredients needed. Comes with some recipes, and others can be added. Requires printer. $49.95 Δ *Autodiff*, a utility that compares two files and produces a report telling how they differ. Requires printer. $49.95. Δ *Eliza*, a variation of the program *Doctor*, which causes the computer to mimic the dialogue of a nondirective psychiatrist. $24.95. Δ *Lisp/86*, an interpreter for the artificial intelligence language Lisp. Includes a simple version of *Eliza* and a guessing game that learns as it goes along. Requires printer. $49.95.

Δ **Micro-Z Applications** (22704 Ventura Boulevard, Woodland Hills, CA 91364; 800-835-2246) has raised the price of its *Millionaire* program to $99.95. ▲

INVENTORY MANAGERS:

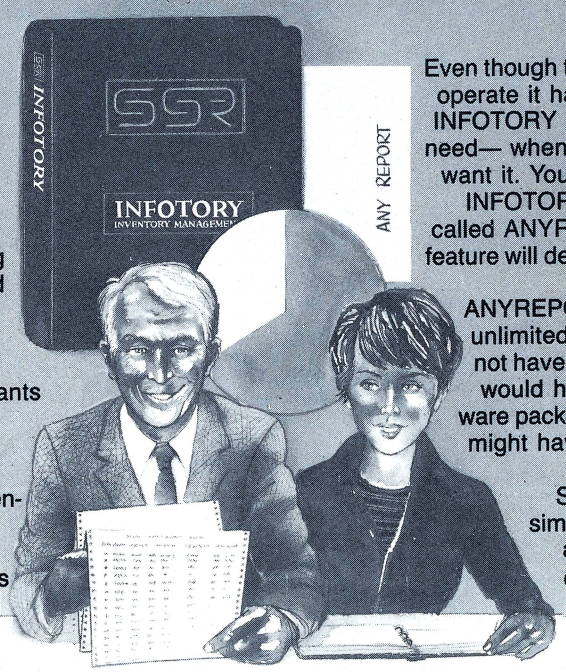
WHEN YOU BUY INFOTORY™... YOU'RE BUYING THE BEST!

SSR Corporation has been providing business solutions for all makes and sizes of computers for over twelve years. They know your needs and how to satisfy them.

They know what management wants in an inventory system...less time learning and more time producing.

Last year SSR introduced INFOTORY...the ultimate solution to inventory management problems.

With INFOTORY *anyone* in the office can learn the system—in less than four hours...because it's so easy to use.



Even though the system is easy to learn and operate it has great flexibility and power. INFOTORY promises the information you need— when you need it *and* the way you want it. You design your own report with INFOTORY'S unique reporting feature called ANYREPORT™. This easy to learn feature will deliver customized reports within minutes.

ANYREPORT's capabilities are almost unlimited. Other inventory systems do not have an ANYREPORT feature. You would have to buy an additional software package to get this extra...and you might have to take a computer course to learn how to use it.

So, for the best, most flexible, simple-to-learn inventory package available...stop in at your nearest IBM PC Dealer, or give SSR a call for more information.

SSR Corporation, 1600 Lyell Avenue, Rochester, New York Telephone 716-254-3200

It's easy.

Easy to use.
Easy to afford.
Now.
The state-of-the-art in spell-checking.
EasySpeller™ from IUS
88,000+ words / 92K disc space
Exclusively, for the IBM personal computer.

EasySpeller™

Extraordinary.

Information Unlimited Software, Inc.
2401 Marinship Way
Sausalito, California 94965
(415) 331-6700

Extraordinary software.

softalk presents the bestsellers

The IBM software marketplace is shaping up as a real horse race, with *WordStar* and *SuperCalc* making determined runs at front-runner *VisiCalc*. The venerable electronic spreadsheet continues to hold forth in the top position, but its dominance is no longer a foregone conclusion.

Two factors enter into the sales figures: First, *SuperCalc* is nipping away at *VisiCalc*'s constituency. For the first time, some stores are reporting Sorcim's product outselling VisiCorp's by slight margins. Likewise, *MBA* is making an impact in the spreadsheet market. Second, *WordStar* has the top end of the word processing market all to itself; and word processing appears to be the second largest application base after business forecasting.

What's becoming apparent is that, at least so far, the IBM Personal Computer is much more a personal business computer than a personal computer. Home and hobbyist software purchases are having little impact on the marketplace and the higher volume dealers are not focusing on the individual user in their sales efforts.

The other side of that coin is that there's relatively little entertainment or home software on the shelves. Infocom has done a yeoman job of delivering its product to the IBM and Epyx/Automated Simulations has speedily adapted *Curse of Ra* to follow *Temple of Apshai*. Continental was ready early with *Home Accountant Plus* and Southeastern Software has converted its *Data Capture 4.0* terminal program.

But the entertainment and home-oriented software that abounds in the Apple, Atari, and TRS-80 marketplaces is still just a gleam in the eye of such software producers as Sirius Software, Sierra On-Line, Synergistic Software, and Southwestern Data Systems. Whether the entry of these companies into the market will result in a broadened home-user base is still to be seen.

Indications are favorable, however. FriendlySoft introduced their *PC Introductory Set*, a package of three game disks and a book, and jumped from nowhere to a tie for fourth on the Top Thirty. Dealers were high on the product, and that reflected in retail sales. To keep all this in perspective, *VisiCalc* and *WordStar* are still selling five to one over the fourth-place product.

Seven other new products also made the Top Thirty in July, all in the lower echelon. *VisiDex* from VisiCorp ranked twenty-fourth in its first month. IBM's *Basic Compiler* was twenty-fifth. *EasyFiler* from Information Unlimited Software tied for twenty-sixth, *Astro Dodge* was twenty-eighth, *Crosstalk* was twenty-ninth, and *Volkswriter* and *Data Capture* tied for thirtieth.

Volkswriter's appearance emphasizes the widespread use of the pc as a word processing machine. The two versions of *EasyWriter*, IBM's and IUS's, are sixteenth and seventh, respectively.

The market seems almost evenly split on accounting packages, with Peachtree's *General Ledger*, as marketed by IBM, continuing to hold an edge over BPI's *General Accounting*. Cyma's small business accounting package maintains a small but regular presence in the market. SSR's *Infotory* is getting some raves also and figures to become a strong contender.

So far, PC-DOS is winning the battle of the operating systems hands-down, with CP/M-86 a narrow second over the p-System. Price appears to be the major consideration, with PC-DOS being the system of choice at the point of purchase and the other two packages being buys at a later date.

The high regard pc users place on communicating with one another is indicated by the presence of *Data Capture* and *Crosstalk* on the Top Thirty. Both pieces of software work in conjunction

We think our new mailing list program is the best in the world. And we're not alone.

1st CLASS MAIL. By Bob Schoenburg and Steve Pollack. Here's a pair of authors who may be software's answer to Irving Wallace. Wallace is the bestselling author who hears the murmuring of general populace and caters to their desires. Bob Schoenburg and Steve Pollack seem to have the same trait.

Consider.

The pair brought out *Home Money Minder*, a perfectly respectable home finance package. Then they listened to the user feedback. The result was *Home Accountant*, one of the phenomenal success stories of the first half of 1982.

Around the same time *Home Money Minder* hit, the team also tested the market with *The Mailroom*. Mailroom never was the success of *HMM*, but the authors used the same technique – listen to the users and incorporate all the good ideas. The result is **1st CLASS MAIL** – a program that, incredibly enough, manages to live up to its double-entendre name.

All of the above is not to accuse Schoenburg and Pollack of putting out the programming equivalent of stalking-horses to do their market research for them. Their original efforts do stand on their own merit. They just pale next to the sequels.

1st CLASS MAIL is so well thought out and so easy to use that other publishers who call their programs "user friendly" should bow their heads in shame.

SOFTALK got an early release of the program sans documentation. Yet a rank computer illiterate was able to apply the program to two separate uses with relatively little trouble. This is high praise indeed: that a novice operator could use a powerful program with no more than the screen menus.

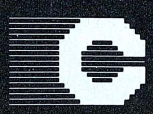
The program allows for twelve fields, clearly more than the traditional name and address of a mailing label. The implication is that the software can be put to other innovative uses as well. The built-in ability to sort and filter on any field or combination thereof enhances the chances that users will find multiple applications for the program.

Continental Software will actually be publishing four versions of the program.* The one already in release is for the Apple II using floppy disks. A hard-disk version will follow. Both versions are pending for Apple III as well, awaiting the development of a rapid binary sort subroutine.

1st CLASS MAIL is a first-rate program for specialized data base applications.

Reviewed by Al Tommervik, Publisher, Softalk.

***Available for: Apple II,™ Apple III™/III with Profile,™ IBM-PC™/IBM-PC with Tecmar™ hard disk/IBM-PC with Davong™ hard disk.**



Continental Software

Continental Software, 11223 South Hindry Avenue Los Angeles, CA 90045 Telephone (213) 417-8031

Copyright 1982 by Softalk Publishing, Inc. Reprinted with permission.

Apple II, Apple III/III with Profile are registered trademarks of Apple Computer, Inc. Tecmar is a registered trademark of Tecmar, Inc. Davong is a registered trademark of Davong Systems, Inc. IBM-PC is a registered trademark of IBM Corp. 1st Class Mail, Home Money Minder and Home Accountant are registered trademarks of Continental Software.

with modems to give pc users easier communication abilities.

Among the programs remaining on the Top Thirty, there was significant juggling of places. *SuperCalc* jumped from sixth to third, Peachtree's *General Ledger* went from fifth to a tie for fourth with *PC Introductory Set*, and BPI's *General Accounting* went from eleventh to sixth. *EasyWriter II* jumped from twenty-fifth to seventh, *IBM Pascal* rose from a tie for twelfth to eighth, and *Zork I* went from a tie for eighth to ninth.

Deadline dropped from fourth to tenth; *Home Accountant Plus* trailed off from third to eleventh. IBM's *Macro Assembler*, still benefiting from sales surges as it becomes available, went from fifteenth to twelfth. Microsoft's *Adventure*, tied for twelfth last month, was thirteenth. Dropping out of the top ten were MicroPro's *MailMerge*, now fourteenth, and IBM's *Asynchronous Communications Support* package, now fifteenth.

EasyWriter 1.1 went from nineteenth to sixteenth. *Temple of Apshai* went from sixteenth to a tie for seventeenth with *Space Guardian*, which had been eighteenth. *Typing Tutor* went from twentieth to nineteenth and *Zork II* went from seventeenth to twentieth.

Holdover programs in the bottom ten are *Total Information Management* and IBM's *Fortran Compiler*, tied for twenty-first; *MBA* at twenty-third; *Accounts Receivable* from Peachtree, tied for twenty-sixth; and *Las Vegas Blackjack*, tied for thirtieth.

Dealers are still finding it easy to sell the pc, most now reporting it as their leading computer; but software sales have generally not kept up. That could be a function of a bottoming-out economy or of the lack of a broad choice of software.

In any event, the softer market in software sales has not affected the three top programs. They remain in demand, and it'll be interesting to see if *VisiCalc* can repulse the upstarts. ▲

IBM Personal Computer sales outlets representing approximately 7.2 percent of all sales of pc and pc-related products volunteered to participate in the poll.

Respondents were contacted early in August to ascertain their sales for the month of July.

The only criterion for inclusion on the list was the number of units sold—such other criteria as quality of product, profitability to the computer store, and personal preference of the individual respondents were not considered.

Respondents in August represented every geographical area of the continental United States.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only for the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus or minus 3.4 percent, which translates roughly into the theoretical possibility of a change of 3.12 points, plus or minus, in any index number.

PC CONNECTION

YOUR LINK TO THE BEST SELLING SOFTWARE FOR THE IBM*PC

SALES LINE 1-800-243-8088

ALASKA, HAWAII, N.H. CALL 1-603-446-3383

Alpha Software	Data Base Manager
	Mailing List
	Question
	Type Faces
	The Apple-IBM Connection
Automated Simulations	Temple of Apshai
	Far Reaches of Apshai
	Curse of Ra
	Jabbertalky
Continental Software	Home Accountant Plus
Infocom	Deadline
	Zork I
	Zork II
Information Unlimited Software	Easy Writer II
	Easy Speller
	Easy Filer
Innovative Software	T.I.M. III
Micro Pro	Wordstar
	Mailmerge
Omric	Space Guardian
Quala	Las Vegas Blackjack
Sorcim	SuperCalc
VisiCorp	VisiCalc
	VisiDex
	Desk Top/PLAN

For More Top Thirty and Other Software for the IBM*PC
Call or Write

PC CONNECTION

PC Connection ■ 6 Mill Street ■ Marlow, N.H. 03456 ■ 1-603-446-3383

Monday through Friday 9:00 to 5:30
VISA and MasterCard Accepted

All Items Subject to Availability
*REGISTERED TRADEMARK OF INTERNATIONAL BUSINESS MACHINES

the top thirty

This Month	Last Month	Index	
1.	1.	257.26	**VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston; VisiCorp, IBM
2.	2.	239.14	**WordStar;** MicroPro
3.	6.	97.83	**SuperCalc;** Sorcim
4.	5.	51.45	**General Ledger;** IBM (Peachtree Software)
	—	51.45	**PC Introductory Set;** FriendlySoft
6.	11.	44.93	**General Accounting,** John Moss and Ken Debower; BPI
7.	25.	42.03	**EasyWriter II,** John Draper; Information Unlimited Software
8.	12.	41.31	**IBM Pascal,** Microsoft; IBM
9.	8.	40.58	**Zork I;** Infocom
10.	4.	39.13	**Deadline;** Infocom
11.	3.	38.41	**Home Accountant Plus,** Mike Farmer, Bob Schoenburg, Larry Grodin, and Steve Pollack; Continental Software
12.	15.	37.68	**Macro Assembler,** Microsoft; IBM
13.	12.	36.96	**Microsoft Adventure,** Gordon Letwin (Will Crowther and Don Woods); IBM (Microsoft)
14.	7.	36.23	**MailMerge;** MicroPro
15.	8.	28.99	**Asynchronous Communications Support;** IBM
16.	19.	28.26	**EasyWriter 1.1,** John Draper; IBM (Information Unlimited Software)
17.	16.	27.54	**Temple of Apshai,** Jeff Johnson, John Freeman, and J.W. Connelly; Epyx/Automated Simulations
	18.	27.54	**Space Guardian;** Omric
19.	20.	26.81	**Typing Tutor,** Michael Sierchio (Dick Ainsworth and Al Baker); IBM (Microsoft)
20.	17.	26.09	**Zork II;** Infocom
21.	10.	22.46	**Total Information Management;** Innovative Software
	12.	22.46	**Fortran Compiler,** Microsoft; IBM
23.	26.	21.74	**MBA;** Context Management Systems
24.	—	18.84	**VisiDex,** Peter Jennings; VisiCorp
25.	—	17.39	**Basic Compiler,** Microsoft; IBM
26.	24.	16.67	**Accounts Receivable;** IBM (Peachtree Software)
	—	16.67	**EasyFiler;** Information Unlimited Software
28.	—	14.49	**Astro Dodge,** Dave Gangola; Digital Marketing
29.	—	13.77	**Crosstalk;** Microstuf
30.	27.	13.04	**Las Vegas Blackjack,** Charles Montante; Quala
	—	13.04	**Volkswriter,** Camilo Wilson; Lifetree Software
	—	13.04	**Data Capture,** David Hughes and George McClelland; Southeastern Software

As IBM would design it

THE PERSONAL HARD DISK™

Only for the IBM Personal Computer

from Corona™

Corona experience makes the difference:

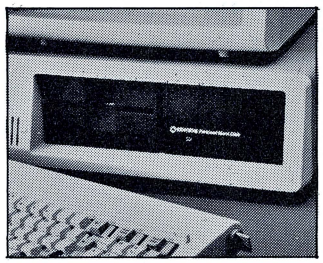
It fits right in—right inside the IBM Personal Computer, with no external power supply.

And it makes friends fast—with self-documenting, menu-driven, very "user-friendly" software.

It responds to your needs—for multiple operating systems on the same disk at the same time, in flexible partitions that grow as needed.

It backs you up—on floppies, using "selective" backup (by date, by volume, by category), so an additional backup device is not needed.

And it has reliability you can count on—combining exclusive use of Seagate drives with Corona's DataGuard™ and FailSafe.™



And the price?

$1995 for 5 MB,

$2495 for 10 MB

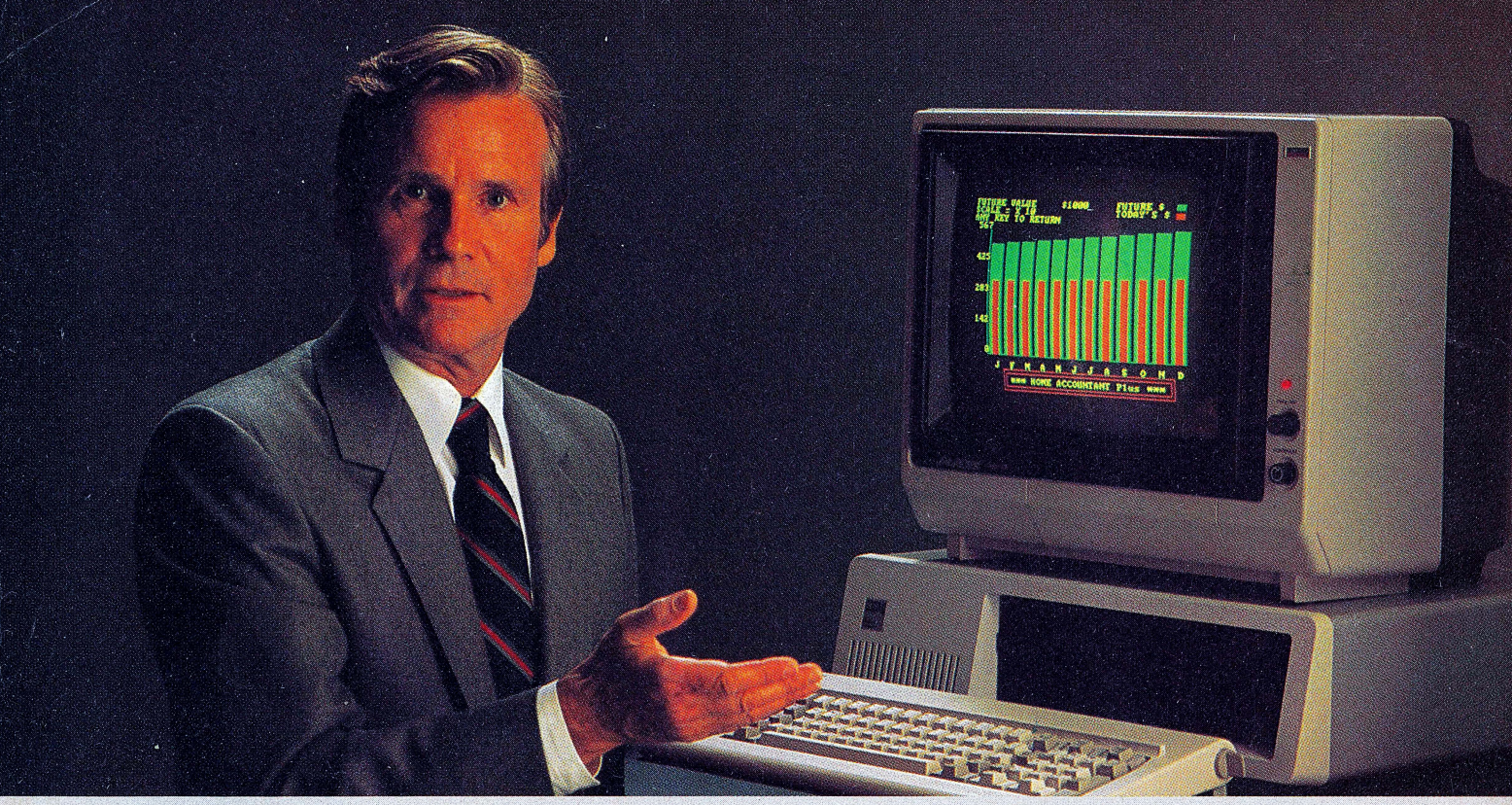
Experience the difference experience makes.

Ask your dealer to show you the Corona Personal Hard Disk (or call us for dealership and quantity purchase information).

corona

The Third Generation Microcomputer Company

Corona Data Systems 31324 Via Colinas, Section 110, Westlake Village, CA 213-706-1505



IBM OWNERS ARE DISCOVERING WHAT OVER 10,000 APPLE OWNERS ALREADY KNOW.

Chances are you don't truly realize how helpful your computer can be in managing your family finances.

The key is selecting the right software. That's why you should know about a program that's become the runaway best seller among Apple owners* — the Home Accountant™ from Continental Software.

ADDED POWER FOR IBM USERS. Home Accountant Plus for the IBM Personal Computer has all the power and ease of operation that made the Apple version such a success. But it includes a brand new Forecasting Module that gives you a peek at the future. You can evaluate various investment strategies — or let the program show the way to your financial goals.

Of course, you get all the great features of Home Accountant. Like room for up to 100 budget categories, 5 different checking accounts, and all the credit cards you can carry. Automatic printing of your net worth and standard financial statements — plus checks and any custom reports that you specify. Also, full-color graphs† of actual vs. budgeted expenditures, trend line analysis, and more.

READY TO GO TO WORK — TODAY. Sooner or later, you're going to want your computer's help with your personal finances.

Over 10,000 users have already found the ideal answer. Now it's your turn.

Home Accountant Plus. About $150 at computer stores everywhere.

Continental Software

Call or write us for your nearest dealer: Continental Software, 11223 Hindry, Los Angeles, CA 90045 (213) 417-8031. *Rated No. 1 in home software packages, Softalk magazine (March-June 1982). †Requires 64k, 1 disk drive and mono or color display. Printer optional. Apple is a registered trademark of Apple Computer Inc. IBM is a registered trademark of IBM.